

# DEPARTMENT OF MECHANICAL ENGINEERING

(NASA-CR-154624-Vol-2) THREE DIMENSIONAL  
THERMAL POLLUTION MODELS. VOLUME 2:  
RIGID-LID MODELS Final Report (Miami Univ.)  
364 p HC A16/MF A01

N80-26751

CSSL 13B

Unclass

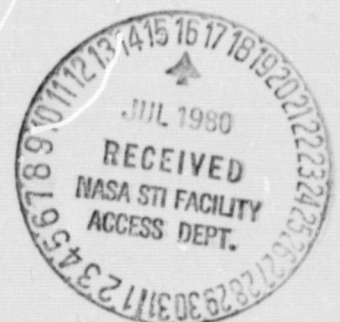
G3/43 33429

## School of Engineering and Architecture

## UNIVERSITY OF MIAMI



**Coral Gables, Florida 33124**



# DEPARTMENT OF MECHANICAL ENGINEERING

(NASA-CR-154624-Vol-2) THREE DIMENSIONAL  
THERMAL POLLUTION MODELS. VOLUME 2:  
RIGID-LIE MODELS Final Report (Miami Univ.)  
364 p HC A16/MF A01

N80-26751

CSCL 13B

Unclass

63/43 33429

## School of Engineering and Architecture

## UNIVERSITY OF MIAMI



Coral Gables, Florida 33124



*Volume II: Rigid-Lid Models*

# **Three Dimensional Thermal Pollution Models**

**NASA Contractor Report CR-154624**

**Contract NAS10-8926  
May 1978**

**Samuel S. Lee and Sabrata Sengupta**

THREE-DIMENSIONAL THERMAL  
POLLUTION MODELS  
VOLUME II - RIGIDLID MODELS

BY

Samuel S. Lee and Subrata Sengupta

Prepared for: /  
National Aeronautics and Space Administration  
(NASA Contract NAS10-8926)

Department of Mechanical Engineering  
School of Engineering and Environmental Design  
University of Miami  
Coral Gables, Florida

May, 1978

72



## (i) PREFACE

This volume is the second of a three volume set presenting description and program documentation of a mathematical model package for thermal pollution analysis and prediction. Two sets of programs one with the rigid-lid formulation and the other with the free-surface formulation were developed by the Thermal Pollution Group at the University of Miami. These models are three-dimensional and time dependent using the primitive equation approach. They have sufficient generality in programming procedure to allow application at sites with diverse topographical features. Both the formulations have near and far field versions. The near field simulating thermal plume areas and the far field version simulating the larger receiving ecosystems. The models simulate the velocity and temperature fields for given meteorological and plant intake and discharge conditions.

The first volume summarizes the mathematical formulation, application experience and overall evaluation of the model package. The present volume, namely Volume II presents the rigid-lid programs. Three versions are presented. One for near field simulation. The second for far-field unstratified situations. The third is for stratified basin far-field simulations. Since these versions have many common subroutines, an unified listing is provided with main programs for three possible application conditions mentioned above. The programs are named NASUM I reflecting NASA funding and technical support and University of Miami effort. The third volume presents the

program documentation for the free-surface models.

These volumes are intended as user's manuals and as such presents specific instructions regarding data preparation for program execution and specific sample problems.

## ACKNOWLEDGEMENTS

The work reported in these three volumes has been the result of co-operation between several institutions and individuals. We wish to acknowledge our sincere gratitude to Mr. Phillip Claybourne, Mr. Reed Barnett and Mr. Roy Bland of NASA-KSC for their continued support during the course of this effort. We are also grateful to Mr. Roy Bland for his technical direction of remote sensing data acquisition and processing. The efforts of Messrs. John Pruitt, Jimmy Neff, Harold Reed, John Renou, Al Bradford and Herb Cribb, were invaluable to the remote sensing studies.

The support of Mr. Charles Dewey, Dr. Ben Sill, Bill McCabe, Jack Jenkins, Bob Edmonds, Ralph Roberts, Miles Majors, Ken Morris, Dave Wiseman, Barry Hurdt and John Gaertner was extremely helpful for the Belews Lake study.

The University of Miami Thermal Pollution Team, in addition to the authors, consisted of

Dr. Norman L. Wienberg

Dr. Homer Hiser

Dr. Josyula Venkata

Mr. Harvey Miller

Dr. Sudarshan Mathavan

Dr. C. F. Tsai

Mr. Ruey Lee

Dr. Cecil Carter

Mr. James Byrne

Various members contributed in different areas of expertise during the course of the study. Special gratitude is expressed to Mr. Harvey Miller for contribution to Volume I and III. Dr. Josyula Venkata was a prime contributor to Volume II.

## LIST OF SYMBOLS

The following list of symbols which are obtained from Volume I are presented here for convenience.

$A_H$	horizontal kinematic eddy viscosity
$A_V$	vertical kinematic eddy viscosity
$A_Z$	vertical eddy viscosity
$A_{ref}$	reference kinematic eddy viscosity
$A_V^*$	$A_V/A_{ref}$
$B_H$	horizontal diffusivity
$B_V$	vertical diffusivity
$B_{ref}$	reference diffusivity
$B_V^*$	$B_V/B_{ref}$
$B_Z$	vertical conductivity : $C_p B_V$
$C_p$	specific heat at constant pressure
$Eu$	Euler number
$f$	Coriolis parameter
$Fr$	Froude Number
$g$	acceleration due to gravity
$h$	depth at any location in the basin
$H$	reference depth
$I$	grid index in x-direction or $x$ direction
$J$	grid index in y-direction or $y$ direction
$K$	grid index in z-direction or $z$ direction
$k$	thermal conductivity

$K_s$	surface heat transfer coefficient
$L$	horizontal length scale
$P$	pressure
$P_s$	surface pressure
$Pr$	turbulent Prandtl number $(\frac{A_{ref}}{B_{ref}})$
$Pe$	Peclet number
$Q^*$	heat sources or sinks
$Re$	Reynolds number (turbulent)
$Ri$	Richardson number
$T$	temperature
$T_{air}$	air temperature
$T_{ref}$	reference temperature
$T_E$	equilibrium temperature
$t$	time
$t_{ref}$	reference time
$u$	velocity in x-direction
$v$	velocity in y-direction
$w$	velocity in z-direction
$x$	horizontal coordinate
$y$	horizontal coordinate
$z$	vertical coordinate

#### Greek Letters

$\alpha$	horizontal coordinate in stretched system
$\beta$	horizontal coordinate in stretched system
$\gamma$	vertical coordinate in stretched system

$\mu$	absolute viscosity
$\rho$	density
$\Phi$	dissipation terms in energy equation
$\tau_{xz}$	surface shear stress in x-direction
$\tau_{yz}$	surface shear stress in y-direction

### Superscripts

(—)	dimensional quantity
( $\sim$ )	dimensional mean quantity
( $'$ )	dimensional fluctuating quantity
( )	dimensional quantity
( ) <sub>ref</sub>	reference quantity

TABLE OF CONTENTS, VOLUME II

i	Preface
ii	List of Symbols for Equations (obtained from Volume I)
1	INTRODUCTION
2	NASUM-I PROGRAM DESCRIPTIONS
2.1	Near-Field
2.11	Description - Algorithm
2.12	Flow Chart (General)
2.2	Far-Field
2.2.1	Description-Algorithm
2.2.2	Flow Chart
3	LIST OF PROGRAM SYMBOLS WITH EXPLANATION IN ALPHABETICAL ORDER (Near Field and Far Field Combined)
4	NASUM-I MAIN PROGRAMS
4.1	Near Field
4.2	Far Field
4.2.1	Unstratified Conditions
4.2.2	Stratified Conditions
5	INPUT DATA
5.1	Near Field
5.1.1	Data Required
5.1.2	Format of Input
5.2	Far Field Model
5.2.1	Data Required
5.2.2	Format of Input
5.2.2.1	For Unstratified Conditions
5.2.2.2	For Stratified Conditions
6	SAMPLE CASE FOR NEAR FIELD
6.1	Problem Statement
6.2	Choice of Programs
6.3	Calculation of Input Parameters



- 6.4 Sample Input
- 6.5 Program Execution Procedure (For constant depth)
- 6.6 Program Execution Procedure (For variable depth)
- 6.7 Sample Output
- 7 SAMPLE CASE FOR FAR-FIELD  
(Unstratified Conditions)
  - 7.1 Problem Statement
  - 7.2 Choice of Programs
  - 7.3 Calculation of Input Parameters
  - 7.4 Sample Input
  - 7.5 Program Execution Procedure
  - 7.6 Sample Output
- 8 SAMPLE CASE FOR FAR-FIELD  
(Stratified Conditions)
  - 8.1 Problem Statement
  - 8.2 Choice of Programs
  - 8.3 Calculation of Input Parameters
  - 8.4 Sample Input
  - 8.5 Program Execution Procedure
  - 8.6 Sample Output
- 9 DESCRIPTION OF COMPUTER PROGRAMS
  - 9.1 Main Programs for the Near-Field and Far-Field
  - 9.2 Subroutines for the Near-Field and Far-Field
- 10 REFERENCES
- 11 APENDICES
  - (a) Equilibrium Temperature and Surface Heat Transfer Coefficient
  - (b) Heat Transfer Mechanisms

## 1.0 INTRODUCTION

This volume presents the main programs and subroutines for NASUM-I as described in Volume I. The program symbols are given in alphabetical order for the convenience of the user, and several sample problems are presented to illustrate clearly what input data is required to execute NASUM-I. Note, that the governing equations and associated approximations and assumptions are given in Volume I.

As previously noted, NASUM-I consists of three main programs: Near-Field Rigid-Lid, Far-Field Rigid-Lid, Unstratified, and Far-Field Rigid-Lid, Stratified. The Near-Field is defined as the domain of interest in the immediate neighborhood of the thermal discharge with open boundaries which extend far enough from the discharge to justify using the Far-Field solutions as boundary conditions on the Near-Field. The Far-Field yields the general circulation and temperature distribution of the water body without the discharge.

The subroutines for NASUM-I are presented in alphabetical order with corresponding program descriptions illustrating to the user an outline of the contents of each subprogram. This allows the user to gain sufficient ease in running NASUM-I, and develop the capability of modifying some of the individual subroutines as desired by the user.

NASUM-I consists of three main programs which use many of the same subroutines, but differ mainly in the use of open boundary conditions for the case of the Near-Field Rigid-Lid Model application. The Far-Field Rigid-Lid Model has been

applied for cases of open boundaries by Sengupta et al (1976), but the nature of the open domain for the Near-Field requires using different main programs which in turn select different subroutines to perform the necessary calculations on these open boundaries.

The Far-Field Rigid-Lid Model programs have been separated into two cases, stratified flow and unstratified flow. This separation is necessary since for deep water bodies seasonal thermoclines are sustained, whereas for shallow water bodies turbulent mixing does not sustain stratification. Thus, for NASUM-I to be of a general nature this separation was essential.

Sample problems, showing sample input and sample output, have been included to provide the user with sufficient background and necessary details involved in executing NASUM-I. Note, that any program modifications should be made with great care, and these modifications should be validated by sample runs to assure the user of the effect these modifications have upon the numerical solutions.

## 2.0 NASUM-I PROGRAM DESCRIPTIONS

The Rigid-Lid Model program contains two parts. The first part is for Near-Field program. The Near-Field here means the region where the effects of the plume on the receiving waters are severe. The second part is for Far-Field. The Far-Field here means the total body of receiving water. The main difference between Rigid-Lid Near-Field and Far-Field lies in the boundary conditions. The Main programs for Near-Field and Far-Field call a different set of subroutines. There will be many subroutines which will be the same for both Near and Far Fields. This is the main reason for combining Near and Far Field programs. The Far-Field main programs had to be divided due to the Computer overflow. The computer used was UNIVAC 1106. In places where there are large computers, the Far-Field main programs can be combined.

### 2.1 Near Field

Near-Field, as said above, is the region where the effects of the discharge are felt most. There are four main programs which are explained in the later sections of this volume. There is no special reason for keeping four main programs and they can be combined. The authors found it convenient to work with four programs since modifications were easier to make during application at various sites.

#### 2.11 Description - Algorithm

The problem is set up as an initial value problem. The values of  $u$ ,  $v$ ,  $w$ ,  $p$ ,  $\rho$ ,  $t$  are assumed known initially.

The values at successive time steps are obtained by using the finite difference equations and marching one step at a time. Fig. 2.1 shows the flow chart. The steps may be summarized as follows:

1. Using the values at time  $n$ , calculate the forcing function in the pressure equation.
2. Solve the pressure equation iteratively.
3. Calculate  $u$  and  $v$  from the  $u$  and  $v$  momentum equations respectively.
4. Calculate  $w$  from the continuity equation.
5. Calculate  $T$  from the energy equation.
6. Calculate  $\rho$  from the equation of state.
7. Check for static stability and apply the infinite mixing criteria for unstable regions.

The values at  $(n+1)$  have thus been obtained. Repeat the above procedure for  $(n+2)$  using values at  $(n+1)$  and so on. The algorithm is essentially the same as that used by Sengupta and Lick (1974).

### 2.12 Flow Chart

Fig. (2.1) shows the general flow chart of the rigid lid programs for both near and far fields. The detailed flow charts with subroutines are given in section 9, where main programs and subroutines are described.

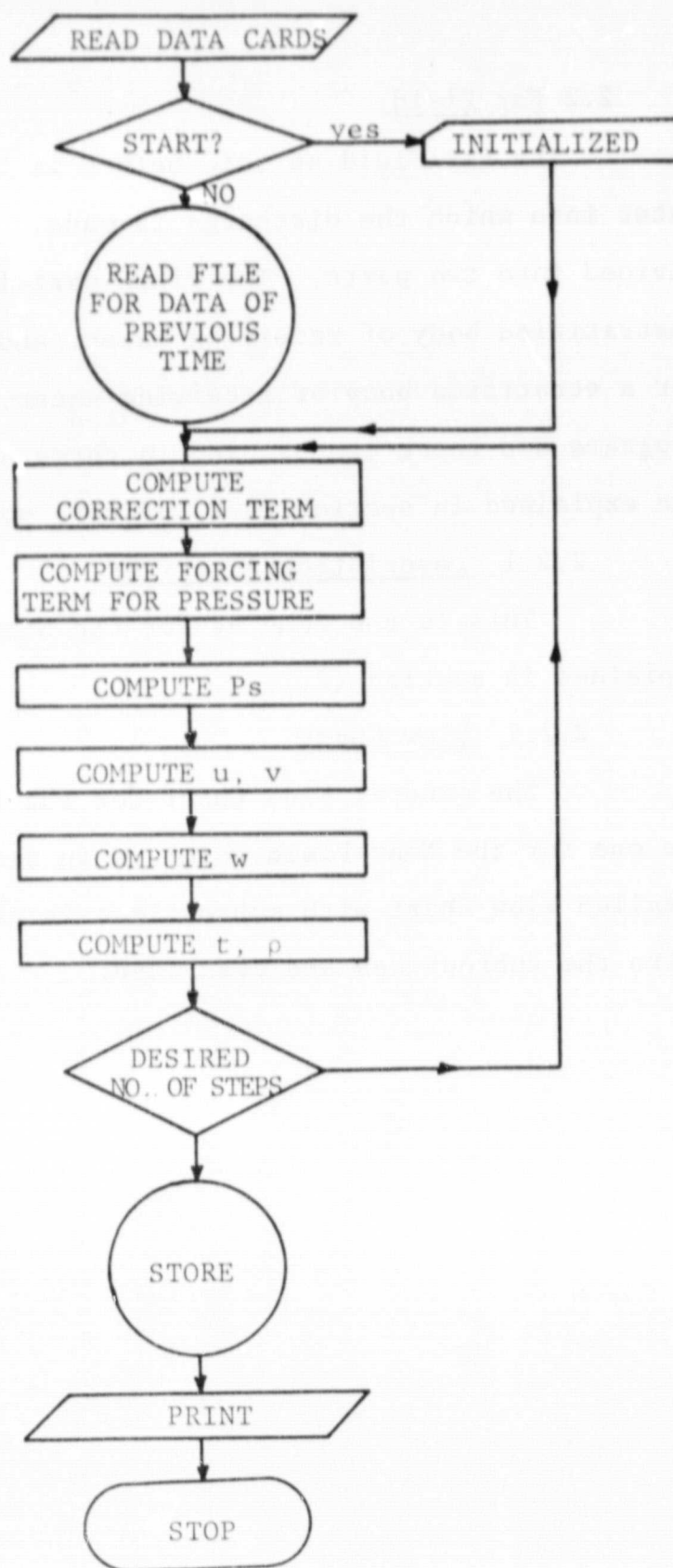


Fig. 2.1 Flow Chart (General)

## 2.2 Far Field

The Far-Field as said before is the total body of water into which the discharge is made. Far-Field programs are divided into two parts. The first part is for the case of an unstratified body of receiving water, and the second part is for a stratified body of receiving water. There are six main programs and these can be used in three different ways which are explained in section (4.2) of this volume.

### 2.2.1 Description-Algorithm

This is the same as for the Near-Field which is explained in section (2.1.1).

### 2.2.2 Flow Chart

The general Flow Chart for the Far-Field is similar to the one for the Near-Field which is in section (2.1.2). A detailed Flow Chart with subroutines is given in section (9.1.5) where the subroutines are described.



### 3.0 LIST OF PROGRAM SYMBOLS WITH EXPLANATION IN ALPHABETICAL ORDER (Near-Field and Far-Field Combined)

## A

A : Constant in Equation of State,  $\rho = A + BT + CT^2$

AA : Value of V at Plume Inlet (ie at I=9) for Near Field

ABR:  $\frac{1}{\text{Rossby Number}}$

AH :  $\frac{1}{\text{Re}}$

AI : Coefficient in front of inertia term

AKT:  $\frac{K_s \text{ Href}}{B_z}$

AP : Coefficient in front of pressure term

ARBP: Arbitrary pressure

AV :  $\frac{1}{\epsilon^2 \text{ RE}}$  where  $\epsilon = \frac{H}{L}$

A3 : Vertical eddy coefficient, normalized with reference viscosity

## B

B : Constant in Equation of State,  $\rho = A + BT + CT^2$

BB : Value of V at Plume inlet (ie at I=10) for near field

BZ :  $\rho C_p B_v$

BV : Vertical eddy diffusion coefficient normalized with reference eddy diffusivity

## C

C : Constant in Equation of State,  $\rho = A + BT + CT^2$

CC : Value for  $\gamma$  constant

CWX: 0.0

CWY: 0.0

D

D : u at previous time step

$$D1TZ: \frac{\partial T}{\partial Z} ; DPX = \frac{\partial P}{\partial X} ; DPY = \frac{\partial P}{\partial Y}$$

$$DPSX: \frac{\partial Ps}{\partial x} ; DPSY = \frac{\partial Ps}{\partial y}$$

DT : Time increment

DX : Increment in x-direction

DY : Increment in y-direction

DZ : Increment in z-direction

$$D1HUX: \frac{\partial (hu)}{\partial x} ; D1HUY = \frac{\partial (hv)}{\partial y} ; D1HUUX = \frac{\partial (huu)}{\partial x} ;$$

$$D1HUVY: \frac{\partial (huv)}{\partial y} ; D1HUVX = \frac{\partial (huv)}{\partial x} ; D1HVVY = \frac{\partial (hvv)}{\partial y}$$

$$D1UY: \frac{\partial u}{\partial y} ; D1VX = \frac{\partial v}{\partial x} ; D2UX = \frac{\partial^2 u}{\partial x^2} ;$$

$$D2VX \frac{\partial^2 v}{\partial x^2} ; D1VWX = \frac{\partial (vw)}{\partial z} ; D1VZ = \frac{\partial u}{\partial z} ;$$

$$D2UZ \frac{\partial^2 u}{\partial z^2} ; D1VZ = \frac{\partial v}{\partial z} ; D2VZ = \frac{\partial^2 v}{\partial z^2} ;$$

$$D1A3Z: \frac{\partial A_3}{\partial z} ; DLZ = \frac{Dx^2 Dy^2}{(Dx^2 + Dy^2)}$$

E

E : v at previous time step

EPS : Convergence criterion

EUL : Euler Number

EX : Residual error

F

FH : Forcing function

G

G : Dummy variable for v

H

H : Dummy variable for u

HI : Non Dimensional depth =  $\frac{h}{H}$

HREF : Reference Depth

HX :  $\frac{\partial H}{\partial \alpha}$  ; HY =  $\frac{\partial H}{\partial \beta}$

I

IN : Maximum number of grid points in x-direction

IWN : Maximum number of half grid points in x-direction,

IWN = IN-1

I : Index for x-axis, Main grid

ITN : Index for number of iterations

IW : Index for x-axis, half grid

IRUN: Index for number of runs

= 0, first run

= 1, from second time onwards

J

J : Index for y-axis, Main grid

JW : Index for y-axis, half grid

JN : Maximum number of main grid points in y-direction

JWN : Maximum number of half grid points in y-direction,

JWN = JN-1

K K : Index for z-axis  
 KN : Maximum number of main grid points in z-direction

L L : Maximum length of the domain  
 LN : Number of time steps to be computed  
 LLN : Total number of time steps/LN

M MAR : Number to describe general location of a point in the  
 main grid.  
 MRH : Number to describe general location of a point in the  
 half grid.  
 MAXIT: Maximum number of iterations

O OMEGA: Relaxation factor

P P : Non-dimensional pressure  
 PN : New pressure, non-dimensional  
 PINH: Dummy variable for pressure

R R : Dimensional density at main grid points  
 RE : Reynolds number  
 RB : Rossby Number  
 RINTX: Density integrated with respect to x  
 RINTY: Density integrated with respect to y  
 RO : Non dimensional density at main grid points  
 ROW : Non dimensional density at half grid points

RREF: Reference density  
 RW : Dimensional density at half grid points  
  
T T : Non dimensional temperature at main grid points  
 TO : Initial temperature (dimensional)  
 TAMB: Ambient temperature (dimensional)  
 TAIR: Air temperature (dimensional)  
 TAI : Coefficient in front of convective terms in the  
         energy equation = 1  
 TAH :  $\frac{1}{Pe}$  where  $Pe = Re \times Pr$   
 TAV :  $\frac{1}{Pe} \epsilon^2$  where  $pe = Re \times Pr; \epsilon = \frac{H}{L}$   
 TE : Equilibrium temperature (dimensional)  
 TTOT: Total time elapsed  
 TAUX:  $\partial u / \partial Y$  (non-dimensional)  
 TAUY:  $\partial v / \partial Y$  (non-dimensional)  
 TEM : Dimensional temperature at main grid points  
 TEMW: Dimensional temperature at half grid points  
 TREF: Reference temperature  
 TW : Non dimensional temperature at half grid points  
 TLL : Temperature at the discharge point (non-dimensional)  
 TMM : Temperature at the discharge point (non-dimensional)  
  
U U : Velocity in x-direction (non-dimensional)  
  
V V : Velocity in y-direction (non-dimensional)  
 VVIS: Vertical eddy viscosity (non-dimensional)



W    W    : Velocity in z-direction (non-dimensional)

WH    : W at half grid points

WHLDT: Time derivative of WH at lid,

ie  $\frac{\partial}{\partial t} (WH)$  at lid

or  $\frac{\partial}{\partial t} (WH)'_{z=0}$

X    XINT: Integral of x terms on the right hand side of  
Poisson's equation. (Eq 2.17, Vol. 1)

X    : Horizontal coordinate across discharge for near field.

Y    YINT: Integral of y terms on the right hand side of  
Poisson's equation. (Eq 2.17, Vol. 1)

Y    : Horizontal coordinate along discharge for near field.

Z    Z    : Vertical coordinate

- W    W    : Velocity in z-direction (non-dimensional)  
       WH   : W at half grid points  
       WHLDT: Time derivative of WH at lid,  
              ie  $\frac{\partial}{\partial t} (WH)$  at lid  
              or  $\frac{\partial}{\partial t} (WH)'_{z=0}$
- X    XINT: Integral of x terms on the right hand side of  
              Poisson's equation. (Eq 2.17, Vol. 1)  
       X    : Horizontal coordinate across discharge for near field.
- Y    YINT: Integral of y terms on the right hand side of  
              Poisson's equation. (Eq 2.17, Vol. 1)  
       Y    : Horizontal coordinate along discharge for near field.
- Z    Z    : Vertical coordinate

4.0 NASUM-I MAIN PROGRAMS : Near Field Main Programs are given for velocity only and for velocity and temperature

#### 4.1 Near Field

There are four main programs for the near field. They are (1) AMAIN 1 (2) AMAIN 2, (3) TMAIN 1 and TMAIN 2. AMAIN programs are to be used when the velocity field is only required, neglecting the effects of density. TMAIN programs are to be used when the velocity and temperature are both required. AMAIN 1 and TMAIN 1 are to be used when the depth is constant. AMAIN 2 and TMAIN 2 are to be used when the depth is variable. All the above main programs can be combined into a single program. But, the authors found it convenient to have four programs to work with.

#### 4.2 Far Field

Far field main programs are provided for unstratified and stratified cooling reservoirs.

##### 4.2.1 Far Field, unstratified

There are six main programs namely, TMAIN 4, TMAIN 4T, TMAIN 5, TMAIN 5T, TMAIN 5V and TMAIN 6. These programs can be used in three different ways. First, if the velocities and temperatures are to be simulated in a coupled fashion, then TMAIN 4, TMAIN 4T, TMAIN 5 and TMAIN 6 are to be used. Second, if the velocities alone are to be simulated, then TMAIN 4, TMAIN 4T, TMAIN 5V and TMAIN 6 are to be used. Third, if the temperatures alone are to be simulated, then TMAIN 4, TMAIN 4T, TMAIN 5T and TMAIN 6 are to be used.



TMAIN 4 and TMAIN 4T initialize the model. TMAIN 6 does the printing of results. Actual simulation is carried out by TMAIN 5, TMAIN 5V and TMAIN 5T. Thus, for successive simulation, TMAIN 5, TMAIN 5V and TMAIN 5T alone need to be executed. TMAIN 6 may be used if results of each run need to be printed.

#### 4.2.2 Far Field Stratified

There are six main programs, namely TMAIN 4CB, TMAIN 4TB, TMAIN 5B, TMAIN 5TB, TMAIN 5VB and TMAIN 6B. These programs can be used in three different ways. First, if the velocities and temperatures are to be simulated in a coupled fashion, then TMAIN 4CB, TMAIN 4TB, TMAIN 5B and TMAIN 6B are to be used. Second, if the velocities alone are to be simulated, then TMAIN 4CB, TMAIN 4TB, TMAIN 5VB and TMAIN 6B are to be used. Third, if the temperatures alone are to be simulated, then TMAIN 4CB, TMAIN 4TB, TMAIN 5TB and TMAIN 6B are to be used.

TMAIN 4CB and TMAIN 4TB initialize the model. TMAIN 6B does the printing of the results. Actual simulation is carried out by TMAIN 5B, TMAIN 5VB and TMAIN 5TB. Thus, for successive simulation, TMAIN 5B, TMAIN 5VB and TMAIN 5TB alone need to be executed. TMAIN 6B may be used if results of each run need to be printed.

## 5.0 Input Data:

The data that is to be given for the execution of the main programs is called Input Data. The data required is explained in section (5.1.1) for the Near field and in section (5.2.1) for the Far-Field. Sections (5.1.2) and (5.2.2) explain the format of input for the near-field and for the far-field respectively. The actual calculation of data for near field and far field are given in the sample problem section to follow.

5.1 Near Field: For near-field there are two classes of programs. They are AMAIN and TMAIN. AMAIN is used for obtaining velocity distribution only. TMAIN is used for velocity and temperature distributions. The data required for running the programs is described in the following section.

5.1.1 Data Required for Near Field: The data required for running TMAIN programs which will simulate velocity and temperature are described below. (The data needed for running AMAIN programs are not given separately as it can be obtained from the data for TMAIN programs). The Fortran symbols which appear in the main program as data are given in the brackets. For the meaning of algebraic symbols the reader is urged to look into the list of symbols.

First time or continuation of run	(IRUN)
Number of cycles	(LN)
Total number of cycles/(LN)	(LLN)
Non-dimensional viscosity (vertical)	(VVIS)
1/Rossby number = $\frac{1}{R_b}$	(ABR)

Coefficient in front of inertia term	(AI)
$1/\text{Reynolds number} = \frac{1}{\text{Re}}$	(AH)
$\frac{1}{\epsilon^2 \text{Re}}$ where $\epsilon = \frac{H}{L}$	(AV)
Coefficient in front of pressure term = 1	(AP)
Convergence criterion	(EPS)
Maximum number of iterations in the solution of Poisson Equation for surface pressure	(MAXIT)
Relaxation factor	(OMEGA)
Arbitrary Pressure for normalizing pressure solution after each iteration	(ARBP)
Grid size in $\alpha$ and $\beta$ directions (Non dimensional)	(DX,DY)
Grid size in $\gamma$ direction (Non-Dimensional)	(DZ)
Coefficient in front of convective terms in the energy equation	(TAI)
$\frac{1}{\text{Peclet Number}} = \frac{1}{\text{Pe}}$ where $\text{Pe} = \text{Re} \times \text{Pr}$	(TAH)
$\frac{1}{\text{Pe} \epsilon^2}$ where $\epsilon = \frac{H}{L}$	(TAV)
Constants in Equation of state	(A,B,C)
(The equation of state is $p = A + BT + CT^2$ where A,B,C are constants, and T is temperature)	
$\frac{K_s H}{B_z}$	(AKT)
where $K_s$ is surface heat transfer coefficient	
$H$ is reference depth and	
$B_z$ is vertical conductivity.	
Euler Number	(EUL)
Temperature gradient at the vertical boundaries	(CW)

Temperature gradient at the bottom	(CB)
Value of v at discharge (for near field at I=9, J=1)	(AA)
Value of v at discharge (for near field at I=10, J=1)	(BB)
Depth of the basin in a constant depth case	(CC)
Discharge temperature at Inlet (for near field at I=9, J=1)	(TLL)
Discharge temperature at Inlet (for near field at I=10, J=1)	(TMM)
Ambient temperature (dimensional)	(TAMB)
$\partial u / \partial y$ $z=0$ in x-direction	(TAUX)
$\partial v / \partial y$ $z=0$ in y-direction	(TAUY)
Time step size	(DT)

Also, depending on the size of the domain the parameter statement has to be changed. In the parameter statement the following parameters are to be changed.

Maximum number of nodes in $\alpha$ -direction, full grid system	(IN)
Maximum number of nodes in $\beta$ -direction, full grid system	(JN)
Maximum number of nodes in $\gamma$ -direction, full grid system	(KN)
Maximum number of nodes in $\alpha$ -direction, half grid system	(IWN)
Maximum number of nodes in $\beta$ -direction, half grid system	(JWN)

#### 5.12 Format of Input For Near Field:

In the previous section (5.11) the data required to run the program is described. In this section, the format in which the data has to be given for the programs to execute will be listed. The calculation of the data is explained in the sample problem section (6.3)

<u>CARD NO.</u>	<u>VARIABLE</u>	<u>FORMAT</u>
1	IRUN	I5
2	LN	"
3	LLN	"
4	VVIS, ABR	FREE
5	AI, AH, AV, AP	"
6	EPS, MAXIT, OMEGA, ARBP	"
7	DX, DY, DZ	"
8	TAI, TAH, TAV	"
9	A, B, C	"
10	TO	"
11	AKT, EUL, CW, CB	"
12	AA, BB, CC	"
13	TLL, TMM	"
14	TAMB	"
15	TAUX, TAUZ	"
16	DT	"

## 5.2 Far-Field

Far-Field main programs are given for stratified and unstratified conditions of the receiving waters. There are six main programs (TMAIN 4, TMAIN 4T, TMAIN5, TMAIN 5T, TMAIN 5V and TMAIN 6) and these programs can be used in three different ways as described before in section (4,2). The data required for these programs is similar to the Near-Field data but it will be repeated again for convenience. Again Fortran symbols that appear in the main programs are shown in brackets.

For Algebraic symbols, the reader is urged to look into the list of symbols.

### 5.2.1 Data Required For Far-Field

Number of cycles for coupled velocity and temperature simulation	(LN)
Number of cycles for uncoupled temperature simulation	(LLN)
Non-dimensional viscosity	(VVIS)
1/Rossby Number	(ABR)
Coefficient in front of inertia-terms	(AI)
$\frac{1}{\text{Reynolds Number}} = \frac{1}{\text{Re}} =$	(AH)
$\frac{1}{\varepsilon^2 \text{ Re}} =$	(AV)
Where $\varepsilon = \frac{H}{L}$	
Coefficient in front of pressure term	(AP)
Convergence criterion	(EPS)
Maximum Number of Iterations in the solution of Poisson Equation for Surface Pressure	(MAXIT)
Relaxation factor	(OMEGA)
Arbitrary Pressure for normalizing pressure solution after each rotation.	(ARBP)
Grid sizes in $\alpha$ and $\beta$ directions, (non-dimensional values)	(DX, DY)
Grid size in $\gamma$ -direction, (non-dimensional value)	(DZ)

Depth of the basin in a constant-depth case, zero otherwise	(CC)
Wind shear in x-direction = $\tau_{xz}$	(TAUX)
Wind shear in y-direction = $\tau_{yz}$	(TAUY)
Time step size, non-dimensional value	(DT)
Coefficient in front of convective terms in the energy equation	(TAI)
$\frac{1}{Pe}$	(TAH)
where $Pe = Re \times Pr$	
$\frac{1}{Pe \epsilon^2}$	(TAV)
where $\epsilon = \frac{H}{L}$ , $Pe = Re \times Pr$	
constants in Equation of state	(A,B,C)
Reference temperature (dimensional)	(TO)
$\frac{K_s H}{B_z}$	(AKT)
Where $K_s$ is surface heat transfer coefficient	
$H$ is reference depth	
$B_z$ is vertical conductivity	
Euler number	(EUL)
Temperature gradient at the vertical boundaries	(CW)
Temperature gradient at the bottom	(CB)
Equilibrium temperature, dimensional	(TAMB)
$\sigma''$ , constant in the equation which is given below for vertical diffusivity in a thermally stratified lake	(CONS)
$A_v = AV_o \{1 + \sigma'' h Tref (-\gamma^2 \frac{\partial \tau}{\partial \gamma})\}^{-1}$	



Depth of the basin in a constant-depth case, zero otherwise	(CC)
Wind shear in x-direction = $\tau_{xz}$	(TAUX)
Wind shear in y-direction = $\tau_{yz}$	(TAUY)
Time step size, non-dimensional value	(DT)
Coefficient in front of convective terms in the energy equation	(TAI)
$\frac{1}{Pe}$	(TAH)
where $Pe = Re \times Pr$	
$\frac{1}{Pe \epsilon^2}$	(TAV)
where $\epsilon = \frac{H}{L}$ , $Pe = Re \times Pr$	
constants in Equation of state	(A,B,C)
Reference temperature (dimensional)	(TO)
$\frac{K_s H}{B_z}$	(AKT)
Where $K_s$ is surface heat transfer coefficient	
$H$ is reference depth	
$B_z$ is vertical conductivity	
Euler number	(EUL)
Temperature gradient at the vertical boundaries	(CW)
Temperature gradient at the bottom	(CB)
Equilibrium temperature, dimensional	(TAMB)
$\sigma''$ , constant in the equation which is given below for vertical diffusivity in a thermally stratified lake	(CONS)
$A_v = A_{v_0} \{1 + \sigma'' h T_{ref} (-\gamma \frac{\partial \tau}{\partial y})\}^{-1}$	



Maximum Value of Vertical diffusivity in a thermally stratified lake, non-dimensional value	(AVMX)
Minimum value of vertical diffusivity in a thermally stratified lake, non-dimensional value	(AVMN)
Identification matrix in the full grid system	(MAR(I,J))
Identification matrix in the half grid system	(MRH(I,J))
Non-dimensional depth matrix	(HI(I,J))
Number of Inlet-Nodes	(NIN)
Number of Outlet Nodes	(NOUT)
Location of Inlet and Outlet Nodes	(I,J,K)
U-velocity at Inlet and Outlet nodes, non-dimensional	(U(I,J,K))
V-velocity at Inlet and Outlet nodes, non-dimensional	(V(I,J,K))
Temperature at Inlet nodes, non-dimensional	
Also initial temperature if available	(T(I,J,K))
Minimum surface temperature in a thermally stratified cooling lake	(TSMN)
Maximum depth over which plume heat is accumulated in a thermally stratified cooling lake	(DPMX)
A matrix of ambient vertical temperature distribution in a stratified cooling lake	(AMINT(N,L))
The first column represents the depths and the second column represents dimensional temperatures	
Number of layers at which ambient temperature is specified. This is the number of rows in AMINT	(NTL)

Number of columns in AMINT (N,L) (NTLV)

Depending on the domain size the following have to be changed in the parameter statement of the main programs

Maximum number of nodes in  $\alpha$ -direction, full grid system (IN)

Maximum number of nodes in  $\beta$ -direction, full grid system (JN)

Maximum number of nodes in  $\gamma$ -direction, full grid system, (KN)

Maximum number of nodes in  $\alpha$ -direction in half grid system (IWN)

Maximum of nodes in  $\beta$ -direction in half grid system (JWN)

## 5.22 Format of Input

In the previous section the data required is described for all the programs in general. In this section, the format in which the data has to be given for each program will be listed. The actual calculation of data will be explained in the sample problem section (7.3) and (8.3) for unstratified and stratified conditions of receiving waters respectively. Since there is more than one main program for each type of ambient condition, the data is given in the form of elements and each element is given a name. The element names and the main programs that goes with it are explained in the next section for the unstratified case and in section (5.2.2.2) for the stratified case.

### 5.2.2.1 Format of Input for Unstratified Conditions:

There are four elements for unstratified type conditions and they are (1) INDATA (2) INDATA 5 (3) INDATA 6 (4) ITPK1 and these go with the main programs (1) TMAIN4, (2) TMAIN5, TMAIN5T and TMAIN5V (3) TMAIN 6 and (4) TMAIN 4T respectively. The data that goes with these data elements is explained in the next sections. The main programs that goes with the elements is given in the brackets.

#### 5.2.2.1.1 Data Element "INDATA" (For main program TMAIN 4)

<u>CARD NO</u>	<u>VARIABLE</u>	<u>FORMAT</u>
1	LN, LLN	16I5
2	VVIS, ABR	Free
3	AI, AH, AV, AP	"
4	EPS, MAXIT, OMEGA, ARBP	"
5	DX, DY, DZ	"
6	CC	"
7	DT	"
8	TAI, TAH, TAV	"
9	A, B, C	"
10	TO	"
11	EUL, DW, EB	"
12	TAMB, AKT, TAUX, TAUY	"
13	MAR (1,1) MAR (2,1) MAR (3,1)...	"
14	MAR (1,2) MAR (2,2) MAR (3,2)...	"
	-----	"
	-----	"
	MRH (1,1), MRH (2,1) MRH (3,1)	"
	MRH (1,2), MRH (2,2) MRH (3,2)	"

<u>VARIABLE</u>	<u>FORMAT</u>
-----	Free
-----	"
HI (1,1) HI (1,2) HI (1,3)	"
HI (1,2) HI (2,2) HI (3,2)	"
-----	
-----	

5.2.2.1.2 Data Element "INDATA 5" (For main programs TMAIN 5,  
TMAIN 5T and TMAIN 5V)

<u>CARD NO</u>	<u>VARIABLE</u>	<u>FORMAT</u>
1	LN, LLN	16I5
2	VVIS, ABR	Free
3	AI, AH, AV, AP	"
4	EPS, MAXIT, OMEGA, ARBP	"
5	DX, DY, DZ	"
6	CC	"
7	DT	"
8	TAI, TAH, TAV	"
9	A,B,C	"
10	TO	"
11	EUL, DW, CB	"
12	TAMB, AKT, TAUX, TAUY	"
13	NIN, NOUT	"
14	I, J, K, U (I, J, K), V(I, J, K), T(I, J, K)	"
	-----	"
	(for inlet)	"
	-----	"
	I, J, K, U(I, J,K), V(I,J,K)	"
	(for inlet)	"
	-----	"

5.2.2.1.3 Data Element INDATA 6 (For main program TMAIN6)

Same as first 12 lines of data element INDATA5.

### 5.2.2.2 Format of Input for Stratified Conditions:

There are four elements for stratified type conditions and they are (1), DATAML, (2) DATAML 5, (3) DATAML 6 and (4) ITLK1 and these data elements go with the main programs (1) TMAIN 4B, (2) TMAIN 5B, TMAIN 5TB and TMAIN 5VB, (3) TMAIN 6B and (4) TMAIN 4TB respectively. The data that goes with these data elements is explained in the next sections. The main programs that goes with the elements is given in the brackets.

#### 5.2.2.2.1 Data Element "DATAML" (for main program TMAIN 4B)

<u>CARD NO</u>	<u>VARIABLE</u>	<u>FORMAT</u>
1	LN, LLN	16I5
2	VVIS, ABR	Free
3	AI, AH, AV, AP	"
4	EPS, MAXIT, OMEGA, ARBP	"
5	DX, DY, DZ	"
6	CC	"
7	DT	"
8	TAI, TAH, TAV	"
9	A,B,C	"
10	TO	"
11	EUL, CW, CB	"
12	TAMB, AKT, TAUX, TAUY	"
13	CONS, AVMX, AVMN	"
14	AMINT (1,1), AMINT (1,2)	"
15	AMINT (2,1), AMINT (2,1)	"
16	AMINT (3,1), AMINT (3,2)	"

<u>VARIABLE</u>	<u>FORMAT</u>
-----	Free
-----	"
MAR (1,1), MAR (2,1) MAR (3,1)...	"
MAR (1,2), MAR (2,2) MAR (3,2)...	"
-----	"
-----	"
MRH (1,1), MRH (2,1), MRH (3,1)...	"
MRH (1,2), MRH (2,2), MRH (3,2)...	"
-----	"
-----	"
HI (1,1), HI (1,2), HI (1,3)...	"
HI (2,1), HI (2,2), HI (2,3)...	"
-----	"
-----	"

5.2.2.2.2 Data Element "DATAML5" (For main programs TMAIN 5B,  
TMAIN 5TB and TMAIN 5VB)

<u>CARD NO</u>	<u>VARIABLE</u>	<u>FORMAT</u>
1	LN, LLN	1615
2	VVIS, ABR	Free
3	AI, AH, AV, AP	"
4	EPS, MAXIT, OMEGA, ARBP	"
5	DX, DY, DZ	"
6	CC	"
7	DT	"
8	TAI, TAH, TAV	"
9	A,B,C	"
10	TO	"

<u>CARD NO.</u>	<u>VARIABLE</u>	34	<u>FORMAT</u>
11	EUL, CW, CB		Free
12	TAMB, AKT, TAUX, TAUY		"
13	CONS, AVMX, AVTN		"
14	NIN, NOUT		"
15	I, J, K, U (I,J,K), V (I,J,K), T(I,J,K)		"
	-----		"
	(for inlet)		"
	-----		"
	I, J, K, U (I, J, K), V (I, J, K)		"
	-----		"
	(for inlet)		"
	-----		"

5.2.2.2.3 Data Element "DATAML6" (For main program TRAIN 6B)

Same as first 13 lines of DATAML 5



## 6.0 Sample Case, Near Field

6.1 Problem Statement: Florida Power and Light Company (FPL) has a fossil fuel power plant situated at Cutler Ridge site on the South Biscayne Bay. The discharge rate is  $1.2 \times 10^7 \text{ cm}^3/\text{sec}$  of water at temperature  $35.9^\circ\text{C}$ . It is required to find three-dimensional velocity and temperature distributions in the near-field (ie in the region where the effects of the thermal discharge are noticeable) for the following environmental conditions.

Air temperature =  $29.5^\circ\text{C}$

Initial temperature of Bay =  $28.0^\circ\text{C}$

Wind speed = 1.2 m/sec, N-W

Average depth of bay = 1.2 m

Discharge width = 10 m

6.2 Choice of Programs: As velocity and temperature distributions are required, and, since depth is constant, the main program to be used is TMAIN 1. If the depth is variable TMAIN2 has to be used.

6.3 Calculation of Input Parameters: In this section, the specification of grid system, reference quantities and discharge velocity chosen will be presented first, followed by the actual calculation of the input data quantities as they appear in the main program.

6.3.1 Grid System: The Remote Sensing data and ground truth data was available for the Cutler Ridge site, (Fig.6.1) and it is used to determine the size of the domain. The data shows that the effects of dis-

charge would be in the region of 500 meters along the discharge axis (y-axis) and 425 meters across the axis of discharge. So, a domain of 500 m X 425 m is selected in the horizontal plane. A grid size of 25 m X 25 m is used. This would give 21 X 18 nodes in the horizontal plane. There are 5 nodes in the vertical direction. This gave a total of 21 X 18 X 5 nodes. The coordinate system and grid system are shown in Figs. (6.2 and 6.3). The MAR and MRH selection are explained in subroutine READ3.

6.3.2 Calculation of Discharge Velocity: The actual discharge width at the site is 10 meters. But, in the numerical model the discharge is 25 meters. So, the discharge velocity is calculated by balancing the mass flow as explained below.

For the numerical problem, with the above grid system, the discharge into the basin would be equal to  $(v \times 25 \times 1.2 \times 10^4) \times 2 \text{ cm}^3/\text{sec}$ .

The actual discharge volume =  $1.2 \times 10^7 \text{ cm}^3/\text{sec}$ .

ie  $(v \times 25 \times 1.2 \times 10^4) \times 2 = 1.2 \times 10^7$

ie  $v = 20 \text{ cm/sec}$

Velocity at discharge or Inlet velocity for the model is 20 cm/sec.

6.3.3 Reference Quantities: Reference Length =  $L$  = Maximum length of domain = 500 meters. Horizontal reference eddy viscosity is calculated using the following formula (Christodoulou et al, 1976)

$$A_{ref} = 0.002 L^{4/3}$$

Where  $L$  is the reference length of the domain in centimeters..

$$A_{ref} = 0.002 (500 \times 100)^{4/3} = 3553.3 \text{ cm}^2/\text{sec}$$

For Near Field problems about 2.8 times the calculated value was found suitable.

Aref = 10,000 cm<sup>2</sup>/sec is used

Vertical reference eddy viscosity = 1 cm<sup>2</sup>/sec

Horizontal reference eddy diffusivity = 10,000 cm<sup>2</sup>/sec

Vertical reference eddy diffusivity = 1 cm<sup>2</sup>/sec

Reference velocity = discharge velocity = 20 cm/sec

Reference temperature = 35.9°C

Reference depth = Average depth of bay = 1.2 meters

#### 6.3.4 Calculation of Input data as it appears in the main program

TMAIN:

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
1	IRUN

IRUN = 0, for the first time and

IRUN = 1, for later runs

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
2	LN

LN can be any number depending on the number of cycles required and the total time the program has to be run. It is always advised to run the program for 10 or 15 cycles and check how the model is running.

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
3	LLN

LLN is Total Number of Cycles

LN

If LN is 15 and LLN is 2, then the model will give output after 15 and 30 cycles.

<u>CARD NO.</u>	<u>FORTRAN STATEMENT</u>
4	VVIS, ABR

$$VVIS = \frac{Av}{Aref} = \frac{1}{10,000} = 0.0001$$

$$ABR = \frac{1}{\text{Rossby Number}} = \frac{1}{Rb} = \frac{fL}{Uref}$$

Where  $f$  is the coriolis function,  $L$  is maximum domain length and  $Uref$  is reference velocity,  $f = 2\Omega \sin \phi$

Where  $\Omega$  is angular velocity of earth,  $\phi$  is the latitude angle.

Since  $Rb$  is large for near field  $ABR \approx 0.0$

<u>CARD NO.</u>	<u>FORTRAN STATEMENT</u>
5	AI, AH, AV, AP

AI is the coefficient in front of inertia term = 1.0

$$AH = \frac{1}{Re} = \frac{Aref}{Uref L} = \frac{10,000}{20 \times 500 \times 100} = \frac{1}{100} = 0.01$$

$$AV = \frac{1}{\epsilon^2 Re} = \frac{L^2}{H^2 Re} = \frac{500 \times 500}{1.2 \times 1.2 \times 100} = 1736.0$$

AP is coefficient in front of pressure term = 1.0

<u>CARD NO.</u>	<u>FORTRAN QUANTITY</u>
6	EPS, MAXIT, OMEGA, ARBP

EPS is the convergence factor and should be set equal to the convergence required. A value of  $EPS = 0.01$  is a good value to start.

MAXIT: is the maximum number of iterations in the solution of Poisson equation for surface pressure. This depends upon the accuracy needed. A value of 50 is a good starting point.

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
7	DX, DY, DZ

$$DX = DY = \frac{\Delta X_{\text{or}} \Delta Y}{L} = \frac{25}{500} = 0.05$$

$$DZ = \frac{1}{(KN-1)} = \frac{1}{4} = 0.25$$

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
8	TAI, TAH, TAV

TAI is the coefficient in front of convective terms in the energy equation = 1

TAH = AH If turbulent Prandtl number is equal to 1.

TAV = AV If turbulent Prandtl number is equal to 1.

$$TAV = \frac{1}{\epsilon^2 Re} = 1736.0$$

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
9	A, B, C

A, B, C are constants in the equation of state and there values are

$$A = 1.000428$$

$$B = -0.000019$$

$$C = -0.0000046$$

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
10	TO

To is the reference temperature = 28.0°C

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
11	AKT, EUL, CW, CB

AKT is non-dimensional heat transfer coefficient =  $\frac{K_s H}{B_z}$

Where  $K_s$  is surface heat transfer coefficient,  $H$  is the reference depth and  $B_z = \rho C_p B_v$

Where  $\rho$  is density,  $C_p$  specific heat at constant pressure,  $B_v$  is vertical diffusivity = 1 cm<sup>2</sup>/sec

$$B_z = 62 \text{ lbm/ft}^3 = 68.05 \text{ slugs/m}^2$$

$$\text{For } K_s = 100 \text{ BTU/}^\circ\text{F} \cdot \text{Ft}^2\text{-day} = 56.5 \text{ cal/}^\circ\text{C} \cdot \text{m}^2 \text{ sec.}$$

$$\text{AKT} = 0.0627$$

$$\text{EUL} = \frac{g H}{U_{\text{ref}}^2} = \frac{980 \times 1.2 \times 100}{20 \times 20} = 294$$

CW and CB are temperature gradient at the vertical boundaries and the bottom respectively and are equal to zero.

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
12	AA, BB, CC

AA, BB are non-dimensional input velocities. Since the discharge velocity is non dimensionalized with reference to discharge velocity AA and BB are equal 1.0 and 1.0.

CC is non dimensionalized depth and equal to 1.0 for a constant depth case.

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
13	TLL, TMM

TLL and TMM are the non-dimensional discharge temperatures.

$$TLL = TMM = \frac{35.9 - 28.0}{28.0} = 0.282$$

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
14	TAMB

TAMB is the air temperature = 29.5°C

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
15	TAUX, TAUY

TUAX, TAUY are the non-dimensional wind shear in x and y directions respectively. Wind shear is obtained from the Wilson (1960) curve which is given in Fig. (6.4). For a wind speed of 1.2 m/sec, the Wilson curve gives the Shear stress ( $\tau_w$ ) equal to 0.049 dynes/cm<sup>2</sup>.

$$\tau_{xz} = \text{Surface shear in x direction} = \tau_w \cos 45^\circ = 0.035$$

$$\tau_{yz} = \text{Surface shear in y direction} = \tau_w \sin 45^\circ = 0.0345$$

$$TAUX = \pm \frac{H}{U_{ref}} \left( \frac{\tau_{xz}}{A_z} \right) = \frac{1.2 \times 100}{20} \times \frac{0.035}{1} = \pm 0.1768$$

$$TAUY = \pm \frac{H}{U_{ref}} \left( \frac{\tau_{yz}}{A_z} \right) = \frac{1.2 \times 100}{20} \times \frac{0.035}{1} = \pm 0.1768$$

The direction +ve or -ve is decided as follows. TAUX (or TAUY) is +ve when the wind force is in the opposite direction of x (or Y) and vice versa. For NW wind, TAUX = +0.1768, TAUY = -0.1768

<u>CARD NO.</u>	<u>FORTRAN QUANTITY</u>
16	DT

In order to determine the time step (DT), the stability analysis has to be made which is done as follows.

$$\text{Convective } \Delta t < \frac{\Delta x}{U} = \frac{25 \times 100}{20} = 125 \text{ sec}$$

$$\text{Viscous } \Delta t < \frac{\Delta x}{2\Delta H} = \frac{(25 \times 100)^2}{2 \times 10,000} = 312 \text{ sec}$$

$$\Delta t < \frac{\Delta x^2}{2\Delta V} = \frac{(0.3 \times 100)^2}{2} = 450 \text{ sec}$$

From above it can be seen that convective criterion is dominant.

ie  $\Delta t < 125 \text{ sec}$

About  $\frac{1}{4}$  times this value is reasonable to use  $\Delta t = \frac{1}{4}(125)$

= 30 sec (approx)

This value of  $\Delta t$  is non-dimensionalized and used as the value for DT.

$$DT = \frac{\Delta T}{t_{\text{ref}}}$$

$$\text{Where } t_{\text{ref}} = \frac{L}{U_{\text{ref}}} = \frac{500 \times 100}{20} = 250 \text{ sec}$$

$$DT = \frac{30}{2500} = 0.01$$



## 6.4 Sample Input

SYMBOLS	VALUES & FORMAT
* 1. IRUN	Ø Ø Ø Ø 0
** 2. LN	Ø Ø Ø 5 0
***3. LLN	Ø Ø Ø Ø 1
4. VVIS, ABR	0.0001, 0.0
5. AI, AH, AV, AP	1.0, 0.01, 1736.1, 1.0
6. EPS, MAXIT, OMEGA, ARBP	0.01, 50, 1.8, 1.0
7. DX, DY, DZ	0.05, 0.05, 0.25
8. TAI, TAH TAV	1.0, 0.01, 1736.1
9. A, B, C	1.000428, -0.000002, -0.0000048
10. TO	28.0
11. AKT, EUL, CW CB	0.627, 294.0, 0.0, 0.0
12. AA, BB, CC	1.0, 1.0, 1.0
13. TLL, TMM	0.282, 0 282
14. TAMB	29.5
15. TAUX, TAUY	0.1768, -0.1768
16. DT	0.01

NOTE: Ø = blank space

\* 0 for the first run and 1 for later runs

\*\* number of cycles required

\*\*\* Prints after 50 cycles in this case. If it is equal to 2, it will print after 50 and 100 cycles.

### 6.5 Program Execution Procedure (For Constant Depth)

In order to execute the programs for the near field constant depth case, the following steps have to be followed.

(1) Input Parameters: The calculation of input parameters is explained in the sample problem section (6.3). The input parameters depend on the discharge conditions, ambient conditions and reference quantities chosen.

(2) First Run: In order to obtain three dimensional velocities and temperatures, the main program to be executed is TMAIN1. In the programs, there are two units, one is read unit designated as, unit 7 and the other is write unit designated as, unit 8. During the first run, there is no need for unit 7, and unit 8 has to be provided to store results. This store unit, '8', would be a magnetic tape.

(3) Continuation of a Run: For extending the results, the run has to be continued. The magnetic tape which was 'unit 8' in the first run would now be read 'unit 7' for reading the previous results. Another magnetic tape is to be provided as unit 8 for storing the extended run results. The above procedure can be repeated until the results are obtained for the desired time. It is to be noted that for the first run  $IRUN=0$  and for continuation of a run  $IRUN=1$ .

The following are a set of control cards that were used on UNIVAC 1106 computer in order to run the near field programs for a constant depth case. The explanation of the control card is given in the brackets.

CARD 1

@ RUN

(Schedule a new run for initiation)

CARD 2

@ ASG,A SKM\*DULL

(All parameters on @ ASG Control Statement are optional except file name.)

A-specifies that the file being assigned is currently catalogued SKM is the qualifier and DULL is file name)

CARD 3

@ PACK SKM'DULL

(Packs the non-deleted elements of a program file, by rewriting the file and eliminating the deleted elements).

CARD 4

@ PREP SKM\*DULL

(Prepares an entry point table for program file, for use by the @ MAP process or in searching a LIB specified program file to satisfy undefined symbols)

CARD 5

@ ASG, T 8., 16N, BAY1

(T-specifies that the file to be assigned temporary and allows it to have a name the same as that of an unassigned catalogued file. BAY1 is the name of the tape.)

CARD 6

@ MAP

(Call the MAP process or (the collector) to collect a specified set of relocatable elements, and produce from this an executable program which is in an absolute element format)

CARD 7

IN SKIM\*DULL. TMAIN1

(TMAIN1 is the main program which would be executed)

CARD 8

LIB SKM\*DULL

(specifies file as a library to be searched)

CARD 9

@ XQT

(Initiates the execution of a program which is in an absolute element format)

CARD 10

Data cards as shown in section (6.4) of the sample problem

— — —

CARD 26

@ FIN

(Terminates a run)

## 6.6 Program Execution Procedure (for variable depth)

In order to execute the programs for the near field variable depth case, the following steps have to be followed.

- (1) Input Parameters: The calculation of input parameters is explained in the sample problem section (6.3). The input parameters depend on the discharge conditions, ambient conditions and reference quantities chosen.
- (2) The depths in the receiving basin at every grid point have to be obtained. If the depth contours of the receiving basin are available, the depths at all grid points can be obtained by interpolation. These depths have to be non-dimensionalized with respect to reference depth. The non-dimensionalized depths have to be read in before execution of the program. In the program, the subroutine HEIGH1 reads the depths, in the direction perpendicular to the discharge axis ie along I or x axis. The depths go as input cards in the format given by subroutine HEIGH1 during the first run after the input card which reads "TAUX, TAUY".
- (3) First Run: In order to obtain three dimensional velocities and temperatures, the main program to be executed is TMAIN 2. In the programs, there are two units. One is read unit designated as, unit 7 and the other is write unit designated as, unit 8. During the first run, there is no need for unit 7, and unit 8 has to be provided to store results. The unit '8', would be a magnetic tape.
- (4) Continuation of a Run: For extending the results, the run has to be continued. The magnetic tape which was used as store unit (unit 8) in the first run would now be the read unit (unit 7) for reading the previous results. Another magnetic tape

has to be provided as unit 8 for storing the extended run results. The above procedure can be repeated until the results are obtained for the desired time period. It is to be noted that for the first run IRUN=0 and for continuation of a run IRUN=1.

The following are a set of control cards that were used on UNIVAC 1106 computer in order to run the near field programs. The explanation of the control cards is given in the brackets.

#### CARD 1

@ RUN

(Schedule a new run for initiation)

#### CARD 2

@ ASG,A SKM\*DULL

(All parameters on @ ASG Control Statement are optional except file name.

A-specifies that the file being assigned is currently catalogued SKM is the qualifier and DULL is file name.)

#### CARD 3

@ PACK SKM\*DULL

(Packs the non-deleted elements of a program file, by rewriting the file and eliminating the deleted elements.)

#### CARD 4

@ PREP SKM\*DULL

(Prepares an entry point table for program file, for use by the a) MAP processor in searching a LIB specified program file to satisfy undefined symbols).

CARD 5

@ ASG,T 8., 16N, BAY1

(T-specifies that the file to be assigned temporary and allows it to have a name the same as that of an unassigned catalogued file. BAY 1 is the name of the tape)

CARD 6

@ MAP

(Call the MAP processor (the collector) to collect a specified set of relocatable elements, and produce from this an executable program which is in an absolute element format)

CARD 7

IN SKM \* DULL TMAIN 1

(TMAIN 1 is the main program which would be executed)

CARD 8

LIB SKM\*DULL

(Specifies file as a library to be searched)

CARD 9

@ XQT

(Initiates the execution of a program which is in an absolute element format)

CARD 10

Data cards as shown in section (6.4) of the sample problem.

— — —

CARD 26

@ ADD SKM\*DULL HEIGH2

(This adds the depths which is stored in the file as an element under the name HEIGH2. This card should be taken out after the first run ie the input cards which read non-dimensionalized depths are included for IRUN=0 only)

CARD 27

@ FIN

(Terminates a run)



### 6.7 Sample Output

The output of the near field rigid lid program consists of the following.

1. Input parameters
2. MAR and MRH matrices
3. Number of iterations (ITN) in the Poissons Equation and residue (EX)
4. Surface Pressure
5. U and V at all grid points (non-dimensional) ie U and V at  $I=1$  to  $IN$ ,  $J=1$  to  $JN$  and  $K=1$  to  $KN$
6. WH at all grid points (non-dimensional) ie WH at  $IN-1$  to  $IWN$ ,  $JW=1$  to  $JWN$  and  $K=1$  to  $KN$
7. Temperatures both non-dimensional and dimensional at all grid points.  
ie T at  $I=1$  to  $IN$ ,  $J=1$  to  $JN$  and  $K=1$  to  $KN$ .

Selected portions of a sample unit are shown in the next few pages.



TYPE 2 EX= .0249403-02

LINE 1

.9712510+00	.9796385+00	.9891320+00	.968368+00	.1020148+01	.9994308+00	.9956555+00	.9972621+00
.9816037+00	.9816037+00	.9795445+00	.9793325+00	.9804468+00	.9825214+00	.9850702+00	.9878784+00
.9803968+00	.9829732+00	.9951650+00	.9973141+00	.9992978+00			

LINE 2

.9721118+00	.9959485+00	.1007276+01	.1017817+01	.1022069+01	.1019648+01	.1013337+01	.1004316+01
.9893115+00	.9891767+00	.9845265+00	.9822157+00	.9817503+00	.9826486+00	.9845964+00	.9870521+00
.9817405+00	.9923119+00	.9947692+00	.9970712+00	.9992371+00			

LINE 1 IF= 1

U-VELOCITY

.0000000	.24322708-01	.1250604+00	.1529122+00	.1872824+00	.2078600+00	.2200564+00	.2246015+00
.2227570+00	.2160208+00	.2062216+00	.1953686+00	.1850261+00	.1764437+00	.1696313+00	.1651219+00
.1619874+00	.1600189+00	.1588789+00	.1583312+00	.1582259+00	.1580205+00		

V-VELOCITY

.0000000	.5099698-01	.6221541-01	.7094028-01	.7949291-01	.8791187-01	.9603377-01	.1035365+00
.1103334+00	.1174145+00	.1233038+00	.1284593+00	.1328277+00	.1364991+00	.1394777+00	.1418059+00
.1438332+00	.1453362+00	.1464228+00	.1472607+00	.1477346+00	.1476875+00		

LINE 1 IF= 2

U-VELOCITY

.0000000	.34322708-01	.1250604+00	.1529122+00	.1872824+00	.2078600+00	.2200564+00	.2246015+00
.2227570+00	.2160208+00	.2062216+00	.1953686+00	.1850261+00	.1764437+00	.1696313+00	.1651219+00
.1619874+00	.1600189+00	.1588789+00	.1583312+00	.1582259+00	.1580205+00		

V-VELOCITY

.0000000	.5099698-01	.6221541-01	.7094028-01	.7949291-01	.8791187-01	.9603377-01	.1035365+00
.1103334+00	.1174145+00	.1233038+00	.1284693+00	.1328277+00	.1364991+00	.1394777+00	.1418059+00
.1438332+00	.1453362+00	.1464228+00	.1472607+00	.1477346+00	.1476875+00		



W= 1 I= 1

WM-VELOCITY

.4396481-01	.1357717-00	.1217223-00	.1202851-00	.1169994-00	.1126212-00	.1074473-00	.1017101-00
.5581072-01	.8722552-01	.7758281-01	.6593132-01	.5578171-01	.4510475-01	.3651339-01	.2909730-01
.2239221-01	.1702783-01	.1130971-01	.6696155-02	-.1791012-02			

W= 1 I= 2

WM-VELOCITY

.6881437-01	.3270424-01	.4115359-01	.3409834-01	.2782663-01	.2446114-01	.2172996-01	.1836613-01
.1723544-01	.1459772-01	.1236063-01	.8092525-02	.6274632-02	.3241206-02	.1785427-02	.7434740-03
.4196624-03	.5023560-03	.2234573-03	.5595127-03	-.3580152-02			

W= 1 I= 1

TEMPERATURE

.3662746-01	.3762813-01	.4110203-01	.4415308-01	.4527573-01	.4361053-01	.3916583-01	.3266981-01
.2527270-01	.1917149-01	.1223506-01	.7833605-02	.4898430-02	.3117855-02	.2126274-02	.1615495-02
.1370770-02	.1261267-02	.1215466-02	.1197704-02	.1191853-02	.1191674-02		

DENSITY

-.2938491-03	-.3020329-03	-.3304221-03	-.3554534-03	-.3646694-03	-.3509508-03	-.3145777-03	-.2616723-03
-.2017042-03	-.1445479-03	-.9705813-04	-.6202592-04	-.3873894-04	-.2463284-04	-.1678930-04	-.1274626-04
-.091492-04	-.9957290-05	-.9593865-05	-.9448523-05	-.9404914-05	-.9404914-05		

W= 1 I= 2

TEMPERATURE

.3662746-01	.3762813-01	.4110203-01	.4415308-01	.4527573-01	.4361053-01	.3916583-01	.3266981-01
.2527270-01	.1917149-01	.1223506-01	.7833605-02	.4898430-02	.3117855-02	.2126274-02	.1615495-02
.1370770-02	.1261267-02	.1215466-02	.1197704-02	.1191853-02	.1191853-02		

DENSITY

-.2938491-03	-.3020329-03	-.3304221-03	-.3554534-03	-.3646694-03	-.3509508-03	-.3145777-03	-.2616723-03
-.2017042-03	-.1445479-03	-.9705813-04	-.6202592-04	-.3873894-04	-.2463284-04	-.1678930-04	-.1274626-04
-.091492-04	-.9957290-05	-.9593865-05	-.9448523-05	-.9404914-05	-.9404914-05		

TEMPERATURES FOR W= 2

29.0	29.1	29.2	29.2	29.2	29.1	28.9	28.7	28.5	28.3	28.2	28.1	28.1	28.0	28.0	28.0	28.0
29.0	29.1	29.2	29.2	29.2	29.1	28.9	28.7	28.5	28.3	28.2	28.1	28.1	28.0	28.0	28.0	28.0
29.3	29.4	29.5	29.7	29.7	29.7	29.0	28.7	28.5	28.3	28.2	28.1	28.1	28.0	28.0	28.0	28.0

ORIGINAL PAGE IS  
OF POOR QUALITY

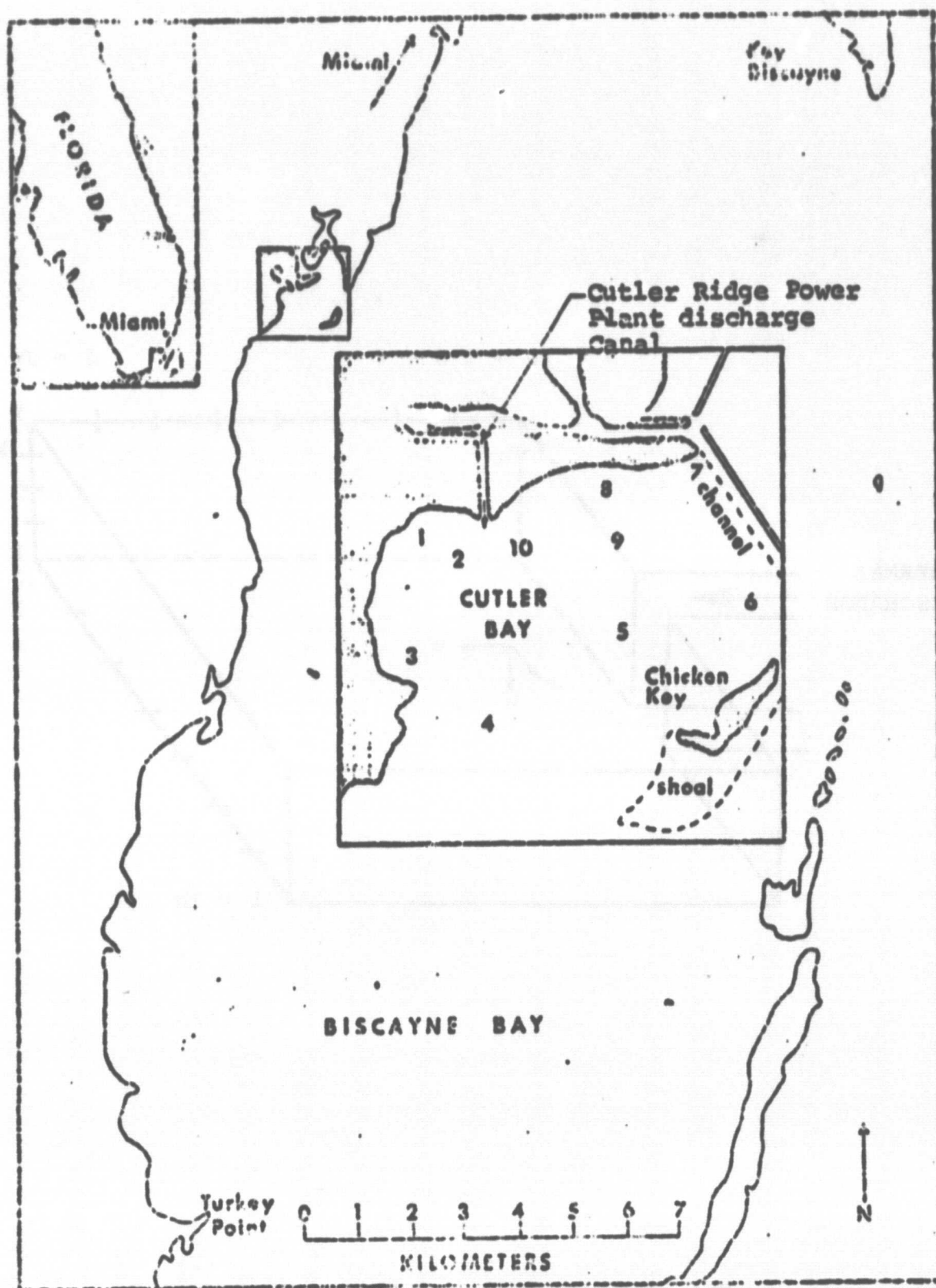


Fig. 6.1 Map showing Cutler Ridge Power Plant and discharge location.



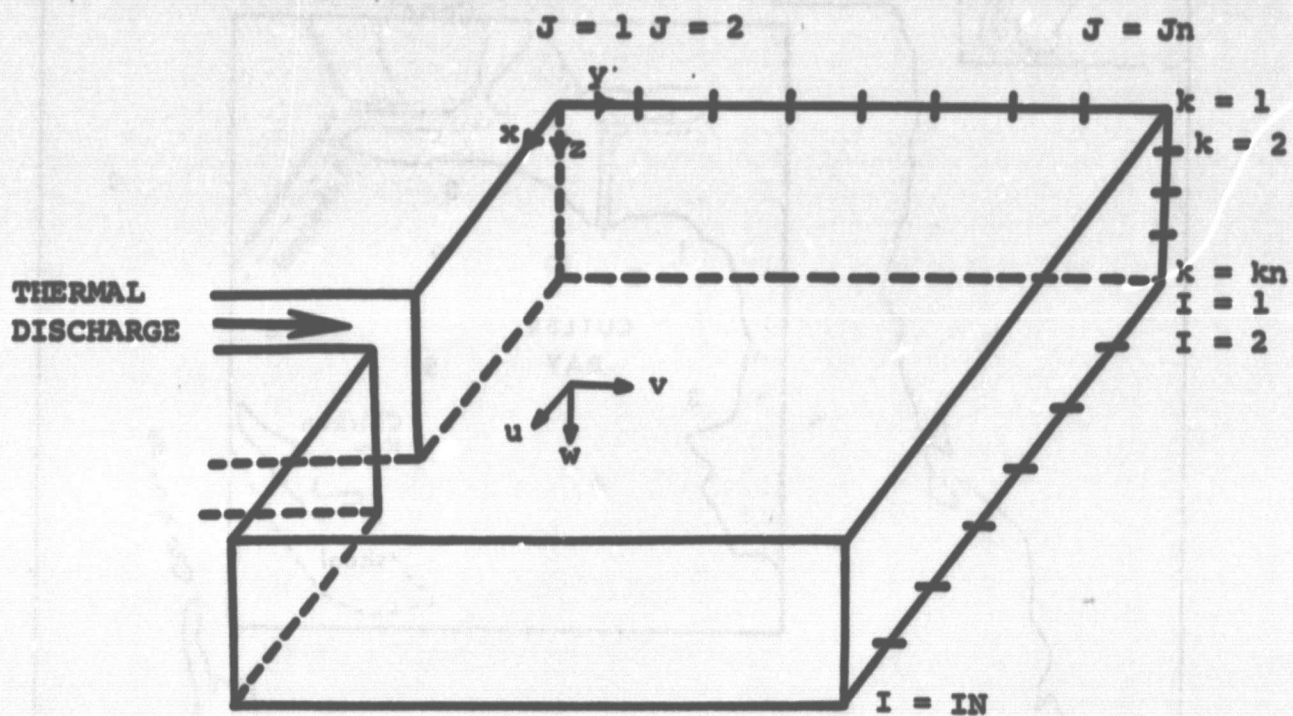


Fig. 6.2 Coordinate and Grid System

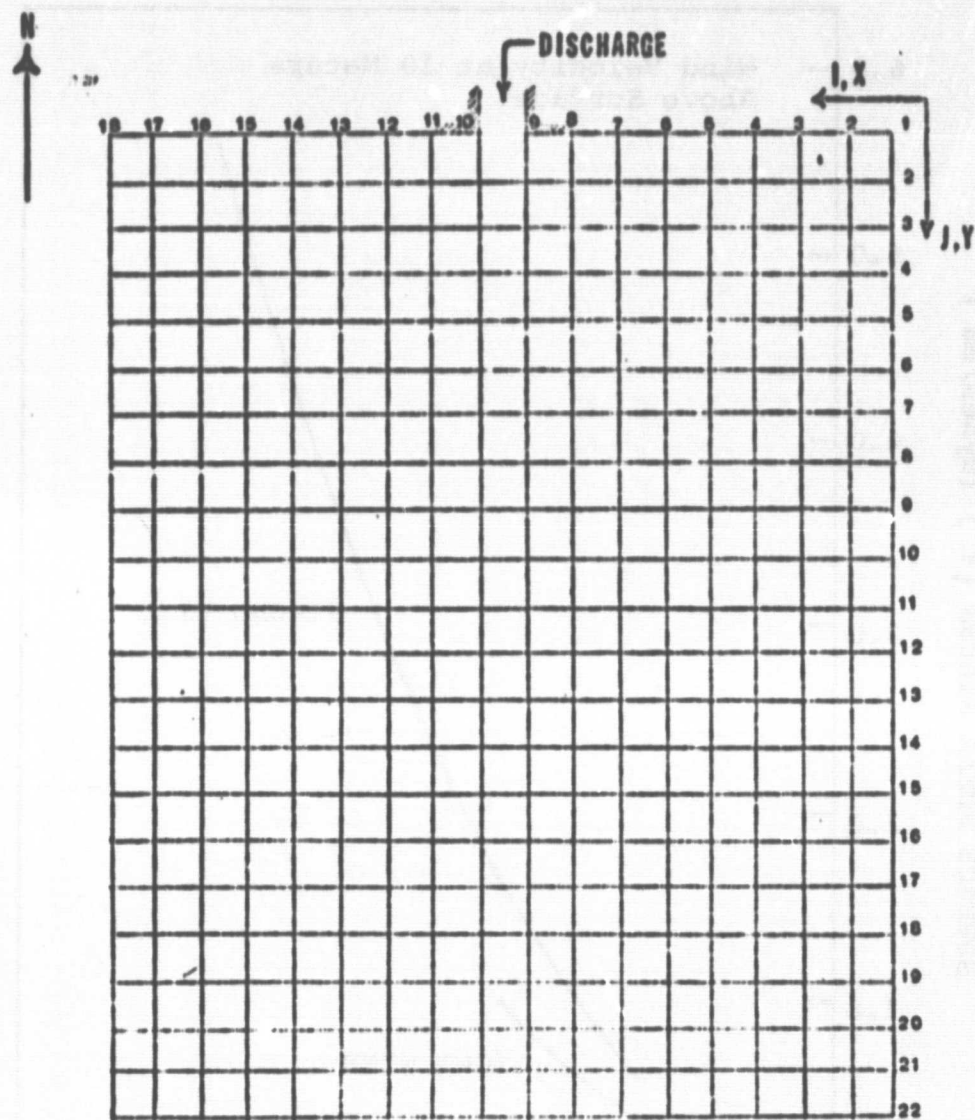


Fig. 6.3 . Grid System For Rigid Lid  
Near Field Model of Cutler  
Ridge Site

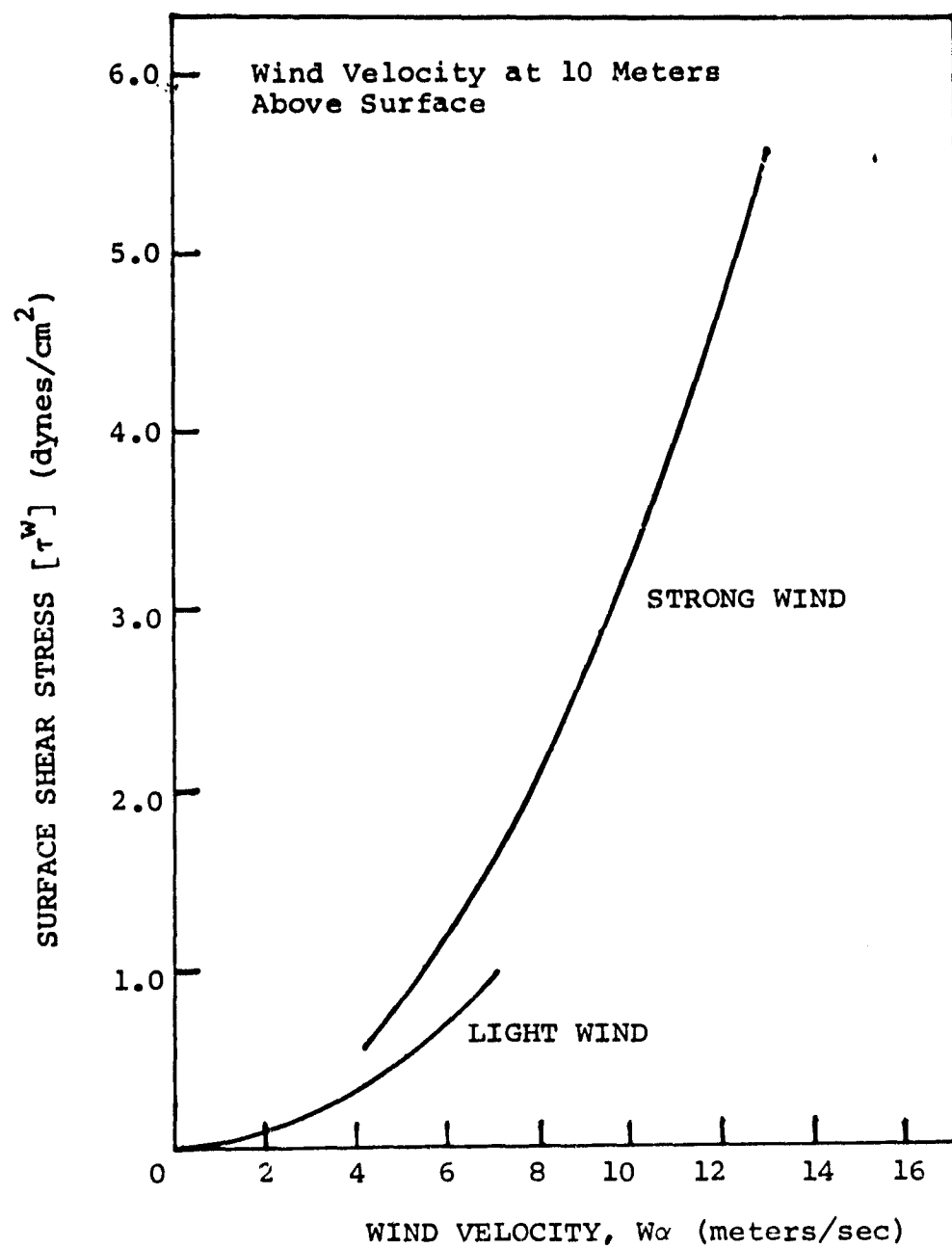


Figure 6.4 WIND SHEAR STRESS RELATION



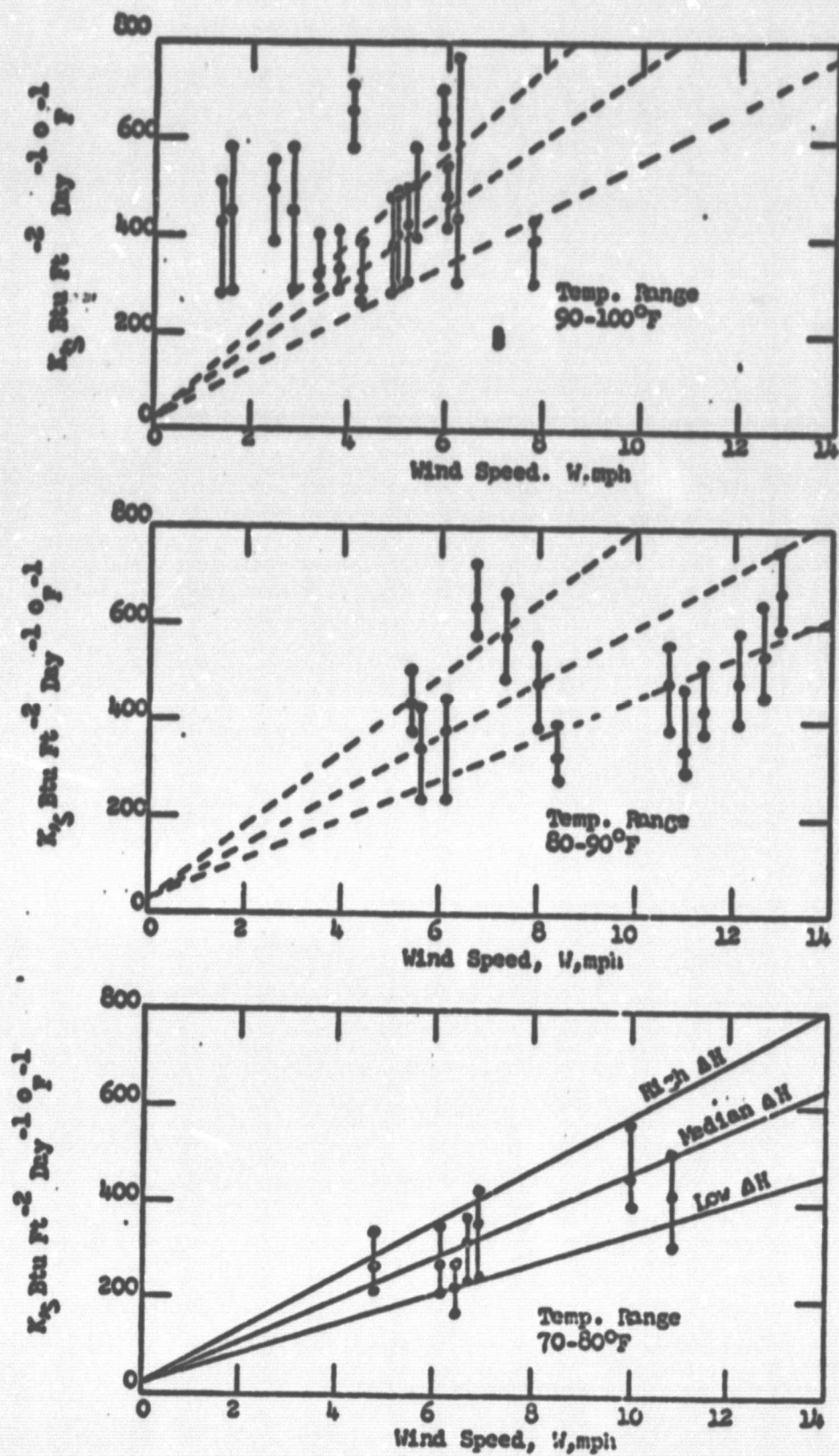


Fig: 6.5 Wind speed relationship

SOME REPRESENTATIVE RESULTS FOR THE  
NEAR-FIELD SAMPLE CASE



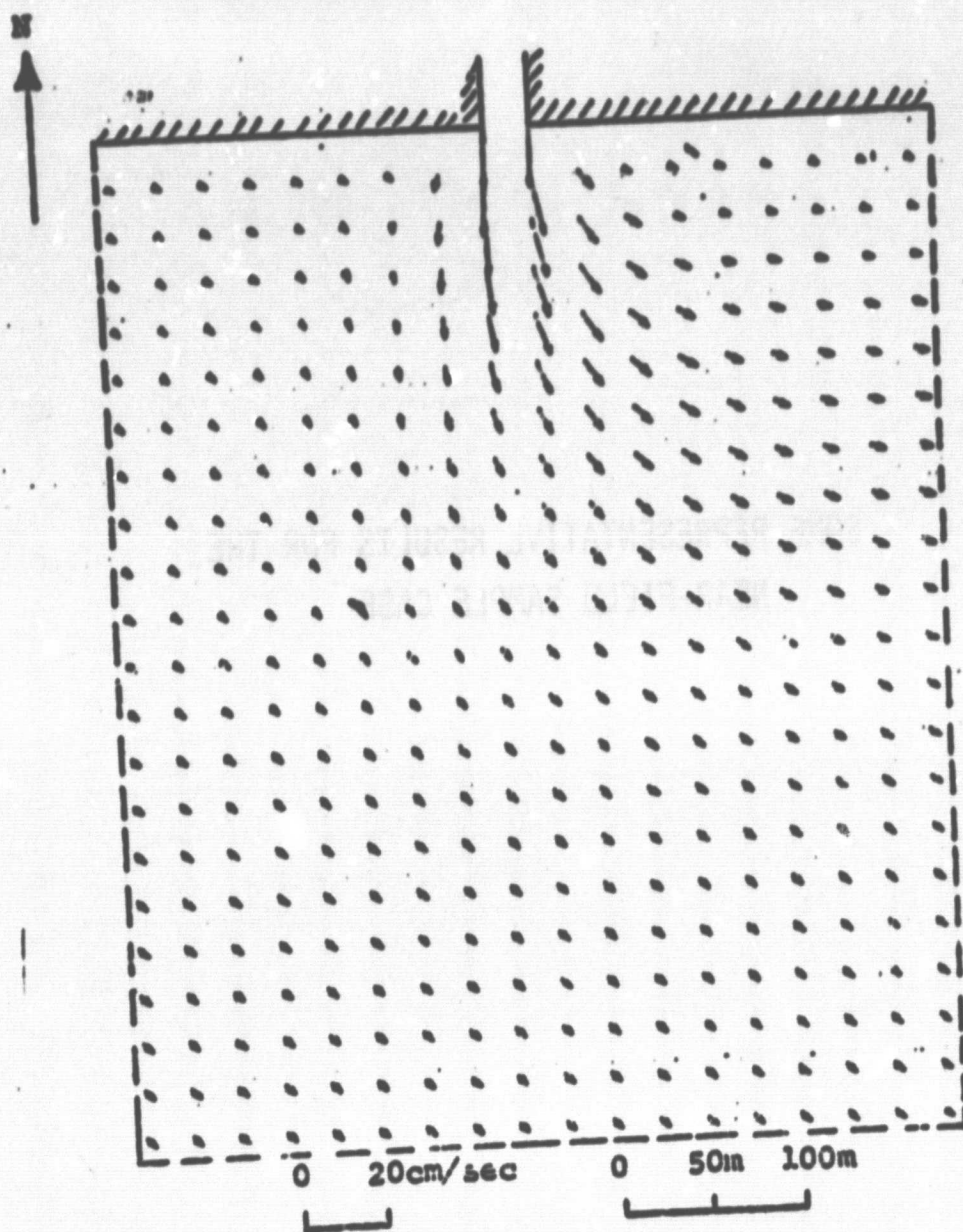


Fig. 6.6 Surface velocity distribution

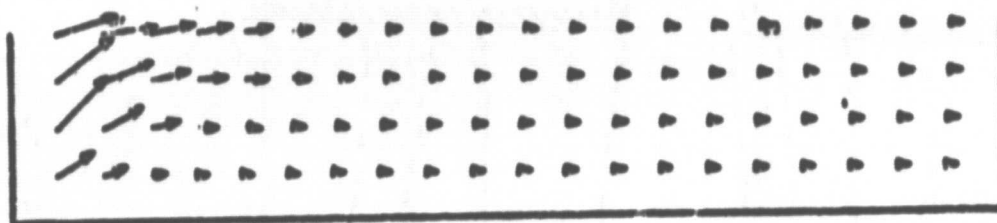


Fig. 6.7 Velocity distribution along the axis of discharge.

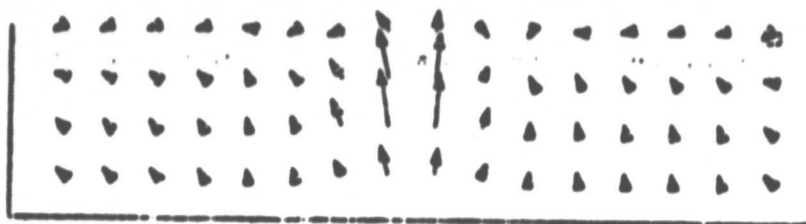


Fig. 6.8 Velocity distribution perpendicular to the axis of discharge.

0 0.0012 cm/sec 0 20 cm/sec

0 0.6m 1.2m

VERTICAL

0 50m 100m

HORIZONTAL

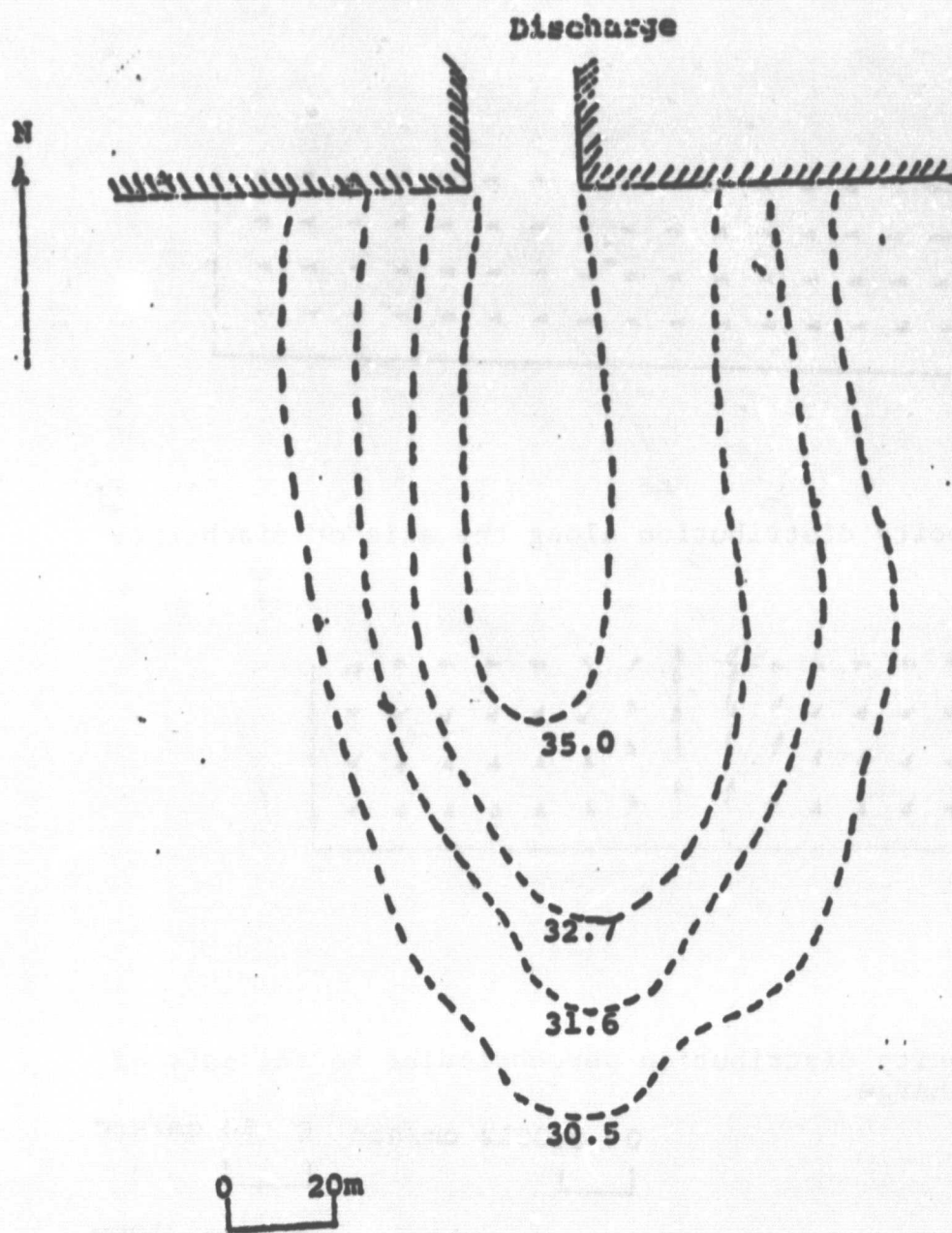
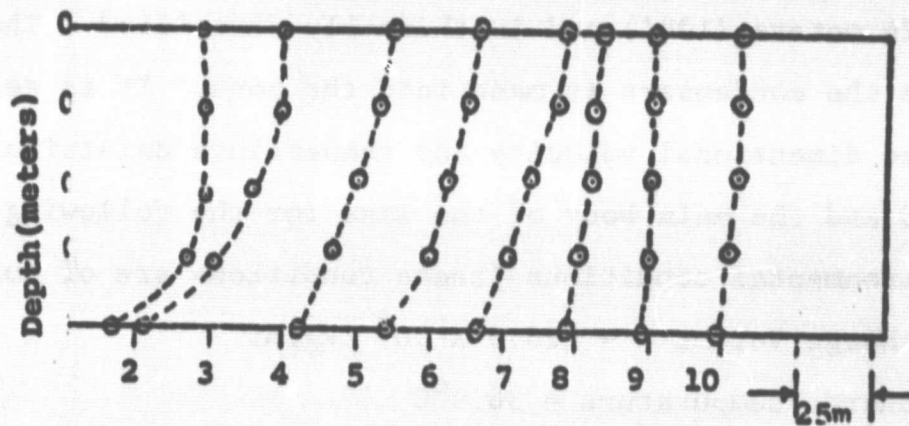


Fig. 6.9 Surface isotherms in the near field.





ance from discharge along Canal center line  
meters.

Fig. 6.10 Vertical section isotherms along canal centerline for the near field.

## 7.0 Sample Case, Far Field (Unstratified Case, Pond)

7.1 Problem Statement: Duke Power Company of North Carolina operates a coal fired electric power plant at Lake Belews Creek in North Carolina. There are two power units and each has a capacity of 1143 MWe. Both these units use Belews Lake as a cooling reservoir. The lake, as shown in figure 7.1 consists of a pond and a main body of the lake. The pond and main body of lake are connected by a canal. The pond has a maximum depth of 13.72 meters (45') and is well mixed. The main body of lake has a maximum depth of 39.26 meters (130') and is thermally stratified. The discharge from the condensers is made into the pond. It is required to find three dimensional velocity and temperature distributions in the pond and the main body of the lake for the following discharge and environmental conditions (these conditions are of August 26, 1976).

Discharge Volume =  $228.9 \times 10^6$  kg/hr  
 Discharge temperature =  $38.9^{\circ}\text{C}$   
 Air temperature =  $25.0^{\circ}\text{C}$   
 Wind speed = 4.25 meters/sec  $225^{\circ}\text{N}$   
 Total incident radiation =  $43.19 \text{ calories/cm}^2/\text{hr}$

7.2 Choice of Programs for Unstratified Pond: Since the pond is unstratified, the main programs to be used are (1) TMAIN 4 (2) TMAIN5, TMAIN 5T, TMAIN 5V and (3) TMAIN 6. The data elements that go with these programs are (1) INDATA (2) INDATA 5 and (3) INDATA 6. If the initial conditions are known (ie Ground truth data and IR data), then the program TMAIN 4T can be used.



The data element that goes with it is IRPKI. TMAIN4T can be eliminated if the initial conditions are not known and one can still obtain velocity and temperature distributions using the model. The calculation of input parameters is explained in the next section.

**7.3 Calculation of Input Parameters:** In this section, the specification of the grid system and reference quantities chosen will be presented first followed by the actual calculation of the input data quantities as they appear in the main programs.

1) **Grid System:** A computational domain of 720 meters by 1680 meters in the horizontal plane as shown in Fig. 7-2 covers

most of the significant portions of the Mixing Pond. The grid size chosen is 60 meters by 60 meters. This gives a 29 X 13 mesh in the horizontal plane. The depth of the pond is 13.72 meters and is divided into five layers thus giving total mesh of 29 X 13 X 6 for the entire computational domain of the pond.

2) **Reference Quantities:** Reference quantities are used to non-dimensionalize the input parameters and they are given below.

Reference Velocity : 30cm/sec

Reference Temperature: 30°C

Reference Horizontal Length = Length of domain = 1680 meters

Reference Depth = Maximum depth of domain = 20 meters

Reference Horizontal eddy viscosity = 45,000 cm<sup>2</sup>/sec

Reference Vertical eddy viscosity = 10 cm<sup>2</sup>/sec

Pr = 1 is chosen

∴ Reference Horizontal eddy diffusivity = 45,000 cm<sup>2</sup>/sec

Reference Vertical eddy diffusivity = 10 cm<sup>2</sup>/sec



### 7.3.1 Calculation of Input data for data element "INDATA" (for main program TMAIN4.)

The definition of the symbols are given in section (5.2.1).

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
1	LN, LLN

These are not calculated values and can be any number depending on the number of cycles required and the total time the program has to be run. It is always advised to run the program for 10 or 15 cycles and check how the model is running.

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
2	VVIS, ABR

$$VVIS = \frac{Av}{Aref} = \frac{1}{45,000}$$

$$ABR = \frac{1}{\text{Rossby Number}} = \frac{1}{RB} = \frac{f L}{Uref}$$

Where  $f$  is coriolis function,  $L$  is domain length and  $Uref$  reference velocity

$$f = 2 \Omega \sin \phi$$

Where  $\Omega$  is angular velocity of earth and  $\phi$  is the latitude angle.

$$f = 2 \times \frac{2\pi}{24 \times 60 \times 60} \sin \phi$$

The latitude of Lake Belews is  $36^{\circ} - 16' - 15''$  North.

This gives  $f = 0.8604 \times 10^{-4}/\text{sec}$

$$\text{Rossby Number} = \frac{Uref}{f L} = \frac{30}{0.8604 \times 10^{-4} \times 1680 \times 100} = 2.07159$$

$$\text{and } ABR = \frac{1}{RB} = \frac{1}{2.07159} = 0.482721$$

CARD NO.

3

FORTTRAN QUANTITY

AI, AH, AV, AP

AI is the coefficient in front of inertia terms in the momentum equations (horizontal) = 1

$$AH = \frac{1}{Re}$$

$$\text{Where } Re = \frac{U_{ref} L}{A_{ref}} = \frac{30 \times 1680 \times 100}{45,000} = 112.2$$

$$AH = \frac{1}{112.2} = 0.008912$$

$$AV = \frac{1}{\epsilon^2 Re}$$

$$\text{Where } \epsilon = \frac{H}{L} = \frac{20}{1680} = 0.0118833$$

$$AV = \frac{1}{(0.0118833)^2 (112.2)} = 63.11$$

AP is the coefficient in front of pressure term and is equal to 1

CARD NO.

4

FORTTRAN QUANTITY

EPS, MAXIT, OMEGA, ARBP

EPS is the convergence criterion and should be set equal to the convergence required. A value of EPS = 0.01 is a good value to start.

MAXIT is the maximum number of iterations in the solution of Poisson Equation for surface pressure. This depends upon the accuracy needed. A value of MAXIT = 50 is a good starting point.

OMEGA is relaxation factor and a value of 1.8 provides rapid convergence.

ARBP is arbitrary pressure and a value of 1.0 is used.



<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
5	DX, DY, DZ

$$DX = \frac{60}{1680} = 0.03565$$

$$DY = \frac{60}{1680} = 0.03565$$

$$DZ = \frac{1}{5} = 0.2$$

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
6	CC

The value of CC = 1.0

(For constant depth CC = 1.0 and zero for variable depth)

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
7	DT

DT is the time step to be used. In order to obtain DT value, stability analysis has to be made in order to see what criterion to be used.

#### Convection Criterion

$$u \frac{\Delta t}{\Delta x} < 1$$

$$\text{or } \Delta t < \frac{\Delta x}{u} = \frac{60 \times 100}{30} = 200 \text{ sec.}$$

#### Viscous Criterion

$$\Delta t < \frac{\Delta x^2}{2A_H} = \frac{60 \times 60 \times 10^4}{2 \times 45,000} = 0.04 \times 10^4 \text{ sec.}$$

$$\Delta t < \frac{\Delta z^2}{2A_V} = \frac{60 \times 60 \times 10^4}{2 \times 10} = 0.2 \times 10^4 \text{ sec.}$$

Diffusive Criterion

$$\Delta t < \frac{\Delta x^2}{2BH}$$

$$\Delta t = \frac{60 \times 60 \times 10^4}{2 \times 45,000}$$

$$= 400 \text{ Sec}$$

The lowest value here is non-dimensionalized with reference to time ( $t_{\text{ref}}$ )

$$t_{\text{ref}} = \frac{L}{U_{\text{ref}}} = \frac{1680 \times 100}{30} = 5.6 \times 10^3 \text{ Sec} = 1.33 \text{ hrs.}$$

The non dimensional time step (DT) =  $\frac{\Delta t}{t_{\text{net}}}$

$$DT = \frac{\Delta t}{t_{\text{ref}}} = \frac{200}{5.6 \times 10^3} = 0.035$$

About  $\frac{1}{2}$  of this value is reasonable to use

$$\therefore DT = \frac{1}{2} (0.035) = 8.92 \times 10^{-3}$$

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
8	TAI, TAH, TAV

TAI is the coefficient in front of convective terms in the energy equation and is equal to 1.

If  $Pr = 1.0$  then  $Aref = Bref$ , since Prndtl number is defined as equal to the ratio of Aref over Bref, and

TAH = AH

TAV = AV

TAI = 1.0

TAH = 0.008912

TAV = 63.11

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
9	A, B, C

A, B, C are coefficients in the equation of state and they are constants. The values are

A = 1.000428

B = -0.000019

C = -0.0000046

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
10	TO

TO is the reference temperature. In this case the reference temperature is taken as 30°C.



<u>CARD NO.</u>	<u>FORTRAN QUANTITY</u>
11	EUL, CW, CB

$$EUL = \frac{gH}{U_{ref}^2} = \frac{980 \times 20 \times 100}{30 \times 30} = 2180$$

CW : Temperature gradient at the vertical boundaries and is equal to zero in this case.

CB : Temperature gradient at the bottom and is equal to zero in this case.

<u>CARD NO.</u>	<u>FORTRAN QUANTITY</u>
12	TAMB, AKT, TAUX, TAUY

TAMB is the equilibrium temperature and is equal to 89.1°F for this case (The details of calculation of equilibrium temperature are given Appendix (A)).

$$AKT = \frac{K_s H}{B_z}$$

Where AKT is non-dimensional surface heat transfer coefficient. Ks is the surface heat transfer coefficient and its evaluation is given in Appendix A.

H is the reference depth and is equal to 20 m

$$B_z = \rho C_p B_v$$

Where  $\rho$  is density,  $C_p$  specific heat at constant pressure,  $B_v$  is vertical diffusivity = 10 cm<sup>2</sup>/sec

For the case considered substituting the above values the value of AKT comes out to be 0.349.

TAUX, TAUY are equal to  $\frac{\partial u}{\partial x}$  and  $\frac{\partial v}{\partial y}$  non-dimensional in x and y directions respectively.

Wind shear is obtained from the Wilson (1960) curve which is given in Fig. (7.5). For a wind speed of 4.25 meters/sec  $225^{\circ}$ N the Wilson curve gives the shear stress ( $\tau_w$ ) equal to  $0.3575 \text{ dynes/cm}^2$ .

$$\tau_w^x = \pm 0.3359 \text{ dynes/cm}^2$$

$$\tau_w^y = \pm 0.1223 \text{ dynes/cm}^2$$

The direction (+ or -) is decided as follows:

$\tau_w^x$  (or  $\tau_w^y$ ) - ve or +ve when wind stress is in the direction of x (or y) or in the opposite direction of x (or y). After finding  $\tau_w^x$  and  $\tau_w^y$  they are non-dimensionalized as shown below to obtain TAUX and TAUy.

$$\text{TAUX} = \frac{H}{U_{\text{ref}}} \frac{\tau_w^x}{A_z} = \frac{10 \times 100}{30} \frac{0.3359}{10} = 2.24$$

$$\text{TAUY} = \frac{H}{U_{\text{ref}}} \frac{\tau_w^y}{A_z} = \frac{10 \times 100}{30} \frac{0.1223}{10} = 0.82$$

<u>CARD NOS..</u>	<u>FORTTRAN QUANTITY</u>
13	MAR(1,1);MAR(2,1):MAR(3,1)

This is to be selected basing on the domain type and an example how to choose is given in Fig. (7.3).

<u>CARD NOS.</u>	<u>FORTTRAN QUANTITY</u>
	MRH (1,1); MRH(2,1): MRH(3,1)

The way MRH is to be selected is similar to MAR. Fig. (7.4)

<u>CARD NOS.</u>	<u>FORTTRAN QUANTITY</u>
	HI(1,1): HI(1,2): HI(1,3)

HI is the non-dimensional depth and changes from one domain to another This is given in subroutine HITEA.

### 7.3.2 Calculation of Input Data For Data Element "INDATA5"

(For main programs TMAIN5, TMAIN 5T and TMAIN5V)

The first 12 input data cards are same as for data element "INDATA" which are calculated in the previous section. Now, it will be shown how to calculate the rest of the data.

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
13	NIN, NOUT

NIN is the number of vertical inlet nodes and is equal to 5 in this case and are at I=2 and J=1. NOUT is the number of vertical outlet nodes and is equal to 5 in this case and are at I=11 and J=13. (See Fig.7.2)

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
14	I,J,K,U(I,J,K),V(I,J,K),T(I,J,K)

(For Inlet)

The discharge or inlet is at I=2, J=1. The discharge is in the y-direction so the velocity in the x direction (v-velocity) is zero and there is only u velocity. The u velocity is calculated as explained below.

The depth at the inlet point is 10.8 meters. There are six grid points in the vertical direction, one being at the surface and one being at the bottom. The velocity at the bottom is zero. The flow depth in the numerical grid system would give  $\frac{1}{2}$  of grid spacing in the vertical direction at the top and bottom. The flow depth associated with intermediate



grid points is one full grid spacing in the vertical direction. The flow width at the inlet is equal to grid spacing in the y direction. Assuming equal velocities at the upper five grid points and zero velocity at the bottom the following formula is used to compute the inlet velocity (U).

$$\text{Velocity (U)} = \frac{\text{Rate of discharge}}{0.9 \times \text{depth} \times \text{width}}$$

$$U = \frac{228.0 \times 10^6 \text{ kg}}{0.9 \times 10.8 \times 60 \times 10^4}$$

$$= 11.9 \text{ cm/sec}$$

Non-dimensionalizing with respect to reference velocity (30 cm/sec) the velocity  $U(I,J,K) = \frac{11.9}{30} = 0.373$ . The discharge temperature is  $38.9^\circ\text{C}$ . Non-dimensional temperature  $T(I,J,K) = \frac{38.9 - 30.0}{30.0} = 0.2967$ .

The input data up to card No. 18 would be the same as card 14 except K value will be increasing by 1

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
19	I,J,K U(I,J,K), V(I,J,K)

(For outlet)

The fluid leaves the mixing pond at location I=11 and J=13 (See Fig. 7.2) This location is called outlet. The outlet velocity is in the y direction and therefore it has only v-velocity. The depth at outlet is equal to depth at inlet. Thus, the velocities at inlet is taken to be equal to velocities at outlet. ie

$$U(I,J,K) = 0.373, V(I,J,K) = 0$$

The rest of the cards (up to 23) would be similar to card 19 except K has to be increased by 1 for every card.

### 7.3.3 Calculation of Input Data For Data Element "INDATA6"

(For Main Program TMAIN6)

Same as first 12 lines of "INDATA" or "INDATA5" which are calculated in the previous sections.

### 7.4 Sample Input (For Far-Field Unstratified Case)

In the previous section the calculation of input parameters for different programs are presented. In this section the calculated numerical values are summarized in order for each Main program.

(Continued)

ORIGINAL PAGE IS  
OF POOR QUALITY



19,0.25:0.50:0.60:0.69:0.69:0.35:0.35:0.35:0.35:1.0:1.0:1.0:1.0  
20,0.19:0.38:0.69:0.69:0.35:0.15:0.15:1.0:1.0:1.0:1.0:1.0:1.0  
21,0.19:0.23:0.53:0.69:0.53:0.23:0.15:1.0:1.0:1.0:1.0:1.0:1.0  
22,1.0:0.23:0.46:0.61:0.69:0.47:0.23:0.19:0.19:0.19:0.19:1.0:1.0  
23,0.16:0.16:0.16:0.47:0.69:0.58:0.26:0.23:0.38:0.38:0.19:1.0:1.0  
24,0.16:0.32:0.53:0.66:0.61:0.69:0.46:0.53:0.47:0.19:0.19:1.0:1.0  
25,0.16:0.16:0.27:0.28:0.46:0.61:0.69:0.64:0.58:0.27:0.27:1.0:1.0  
26,1.0:1.0:1.0:0.21:0.41:0.50:0.61:0.69:0.69:0.53:0.27:1.0:1.0  
27,1.0:1.0:1.0:0.21:0.21:0.25:0.50:0.61:0.69:0.53:0.27:0.27:1.0  
28,1.0:1.0:1.0:1.0:1.0:0.20:0.38:0.53:0.53:0.53:0.38:0.27:1.0  
29,1.0:1.0:1.0:1.0:1.0:0.27:0.27:0.27:0.27:0.27:0.27:0.27:1.0

# 7.4.2 Sample Input for Main Programs TMAIN 5, TMAIN 5T and TMAIN 5V (IN DATA 5)

```

1      20  23
2      0.0003222,0.4827
3      1.0,0.008912,63.117,1.0
4      0.01,100,1.3,1.0
5      0.03565,0.03565,3.20
6      0.0
7      0.00107
8      1.0,0.008912,63.117
9      1.000428,-0.000019,-0.0000046
10     30.0
11     2180.0,0.0,0.0
12     31.7,0.349,0.896,0.328
13     5,5
14     2,1,1,0.0,0.373,0.2967
15     2,1,2,0.0,0.373,0.2967
16     2,1,3,0.0,0.373,0.2967
17     2,1,4,0.0,0.373,0.2967
18     2,1,5,0.0,0.373,0.2967
19     11,13,1,0.0,0.373
20     11,13,2,0.0,0.373
21     11,13,3,0.0,0.373
22     11,13,4,0.0,0.373
23     11,13,5,0.0,0.373

```

## 7.4.3 Sample Input for TMAIN 6 (IN DATA 6)

```

1      20      20
2      0.0002222,0.4827
3      1.0,0.006912,61.117,1.0
4      0.01,100,1.8,1.0
5      0.03165,0.03505,0.20
6      0.0
7      0.00107
8      1.0,0.006912,61.117
9      1.000426,-0.0000019,-0.0000046
10     30.0
11     2180.0,0.0,0.0
12     11.7,0.349,0.896,0.328

```



### 7.5 PROGRAM EXECUTION PROCEDURE:

In order to execute the programs for the far field model, the following steps have to be followed.

1) Input Parameters: The calculation of input parameters is explained in the sample problem section (7.3). The input parameters depend on the discharge conditions, ambient conditions and the reference quantities chosen.

2) First Run: In order to obtain three dimensional velocities and temperatures, the main programs that have to be executed are TMAIN 4, TMAIN 4T, TMAIN 5 and TMAIN 6. The flow chart is shown in Figure (7.6a). The main program TMAIN 4 initializes velocities and temperatures ie it sets, all velocities equal to zero and temperature equal to the reference temperature. TMAIN 4T reads IR data as the initial conditons for temperature. If IR data is not available, the main program TMAIN4T should not be executed. TMAIN5 does computations and TMAIN 6 prints the results. In the programs, there are two units. One is read unit, designated as unit 7 and the other is store unit, designated as unit 8. Two tapes have to be provided, one for the read unit (unit 7) and another for the store unit (unit 8).

3) Continuation of a Run: For extending the previous results for more time, the run has to be continued. Now the programs that need to be executed are TMAIN 5 and TMAIN 6. Two tapes have to be provided for the continuation of a run also.

The following are a set of a control cards that were used on UNIVAC 1106 computer in order to run the programs for the first time for a far-field unstratified receiving basin. The explanation for the control cards is given in

the brackets.

#### CARD 1

@ RUN

(Schedule a new run for initiation)

#### CARD 2

@ASG,A SKM\*DULL

(All parameters on @ ASG Control Statement are optional except file name.

A-specifies that the file being assigned is currently catalogued. SKM is the qualifier and DULL is file name.)

#### CARD 3

@PACK SKM \* DULL

(Packs the non-deleted elements of a program file, by rewriting the file and eliminating the deleted elements)

#### CARD 4

@ PREP SKM \* DULL

(Prepares an entry point table for program file, for use by the @ MAP processor in searching a LIB specified program file to satisfy undefined symbols)

#### CARD 5

@ ASG,T 8., 16N, LAKE 1

(T-specifies that the file to be assigned temporary and allows it to have a name the same as that of an unassigned catalogued



file. LAKE 1 is name of the tape)

#### CARD 6

@ MAP

(Call the MAP processor (the collector) to collect a specified set of relocatable elements, and produce from this and executable program which is in an absolute element format)

#### CARD 7

IN SKM \* DULL TMAIN 4

(TMAIN 1 is the main program which would be executed)

#### CARD 8

LIB SKM \* DULL

(Specifies file as a library to be searched)

#### CARD 9

@ XQT

(Initiates the execution of a program which is in an absolute element format)

#### CARD 10

@ ADD SKM \* DULL INDATA

(Note: INDATA is the data element that provides the input data for the TMAIN 4 main program. The calculation of INDATA is given in the sample problem)

CARD 11

@ ASG,T 7., 16N, LAKE 1

CARD 12

@ ASG,T 8., 16N, LAKE 2

(Note: LAKE 2 is the name of the tape)

CARD 13

@ MAP

CARD 14

IN SKM \* DULL TMAIN 4T

CARD 15

LIB SKM\*DULL

CARD 16

@ XQT

CARD 17

@ ADD SKM & DULL ITPK1

(Note: ITPK1 is the data element that provides the input data for the TMAIN 4T main program. The calculation of ITPK1 is given in the sample problem)

CARD 18

@ ASG,T 7., 16N, LAKE 2

CARD 19

@ ASG,T      8., 16N, LAKE1

CARD 20

@ MAP

CARD 21

IN SKM \*DULL TMAIN5

CARD 22

LIB SKM \* DULL

CARD 23

@ XQT

CARD 24

@ ADD SKM \* DULL INDATA5

(Note: INDATA 5 is the data element that provides the input data for the TMAIN 5 main program. The calculation of INDATA5 is given in the sample problem)

CARD 25

@ ASG,T      7., 16N. LAKE1

CARD 26

@ MAP

CARD 27

IN SKM \* DULL TMAIN6

CARD 28

LIB SKM \* DULL

CARD 29

@ XQT

CARD 30

@ ADD SKM \* DULL INDATA 6

(Note: INDATA6 is the data element that provides the input data for the TMAIN 6 main program. The calculation of INDATA6 is given in the sample problem.)



## 7.6 Sample Output

The sample output for far field is similar to the near field, which is given in section 6.6.



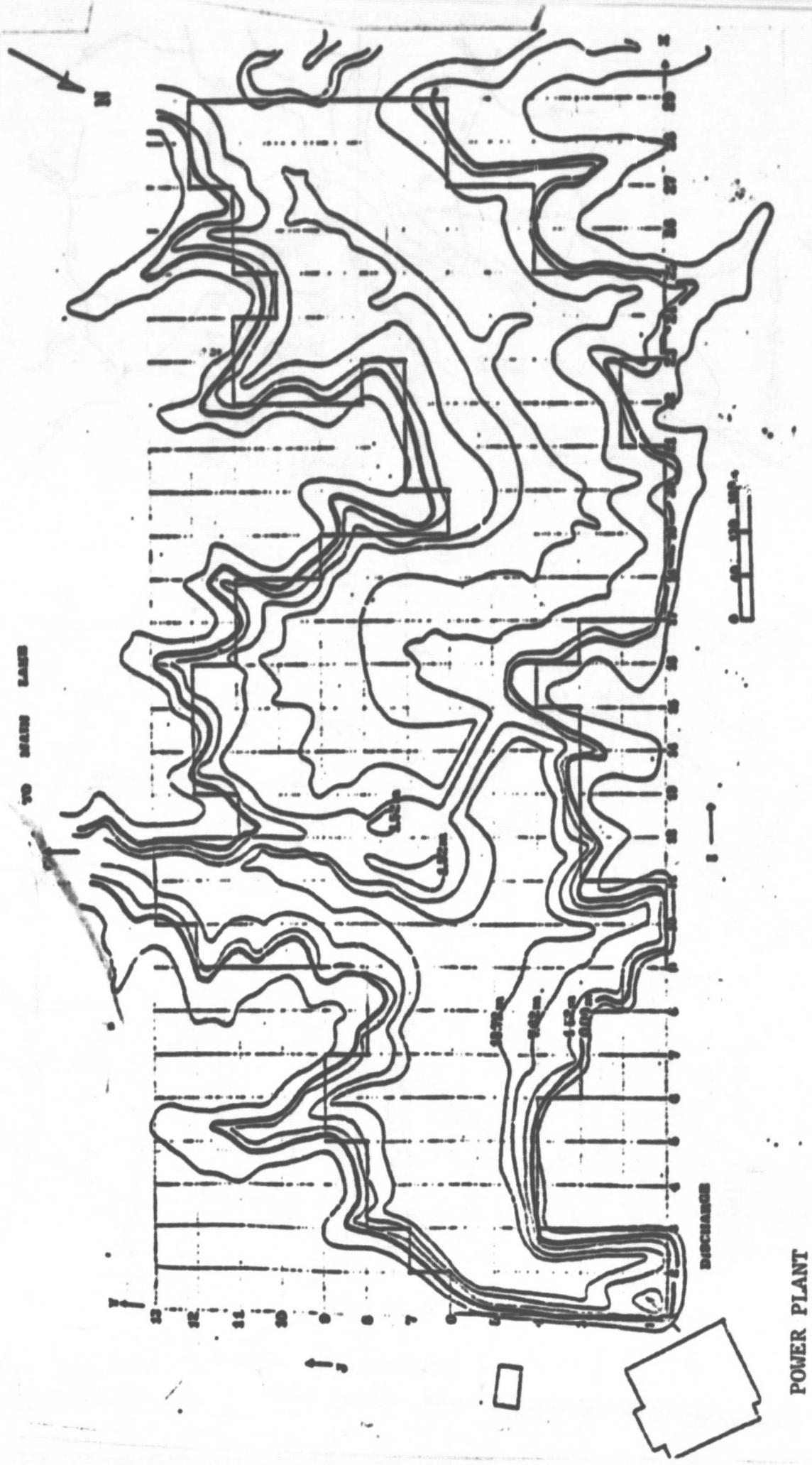


Fig. 7.2 Computational grid for mixing pond at Lake Belevs site.

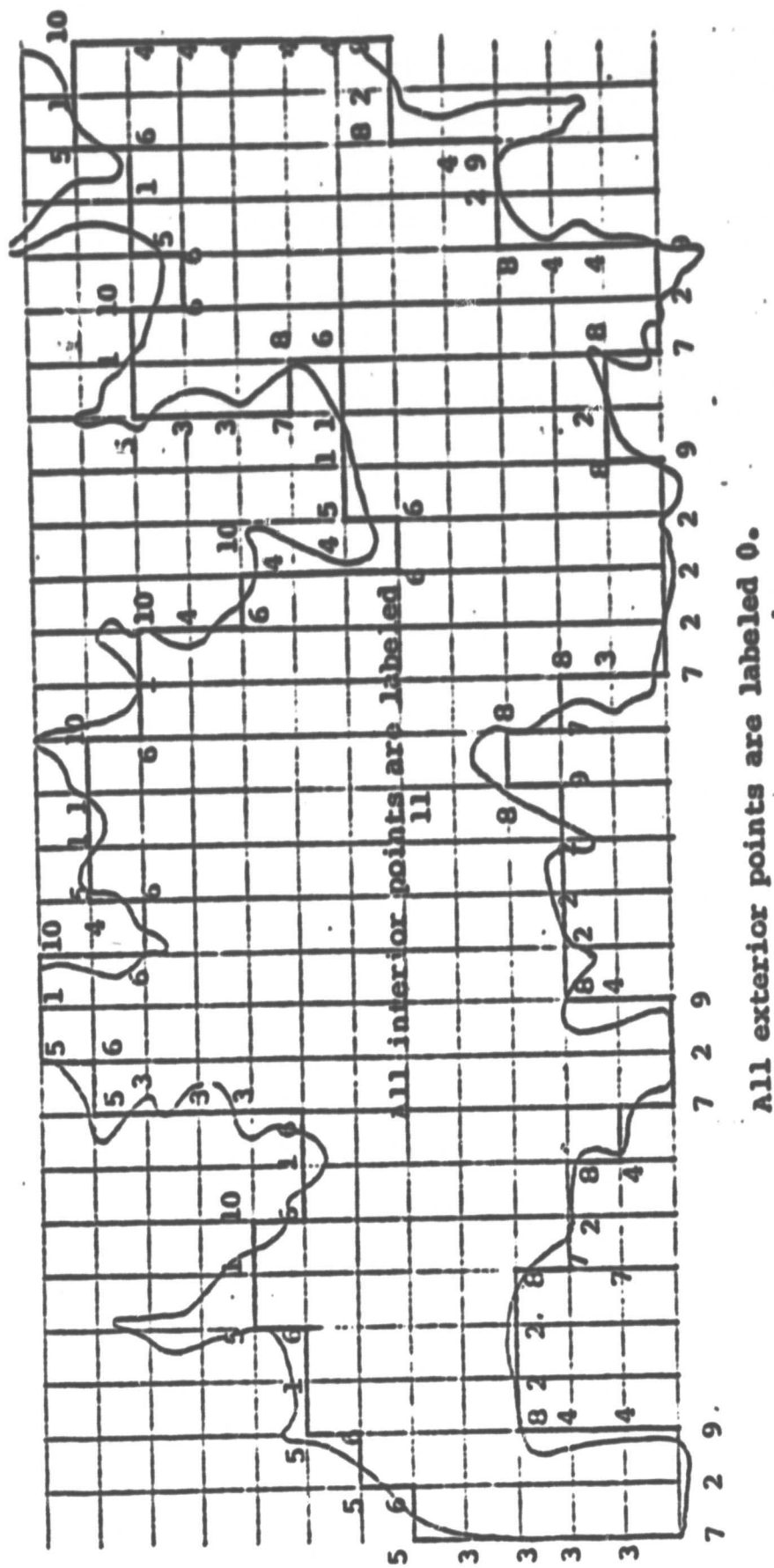


FIG. 7.3 Identifying numbers in the Main Grid System.  
the MAR matrix for the Mixing Pond



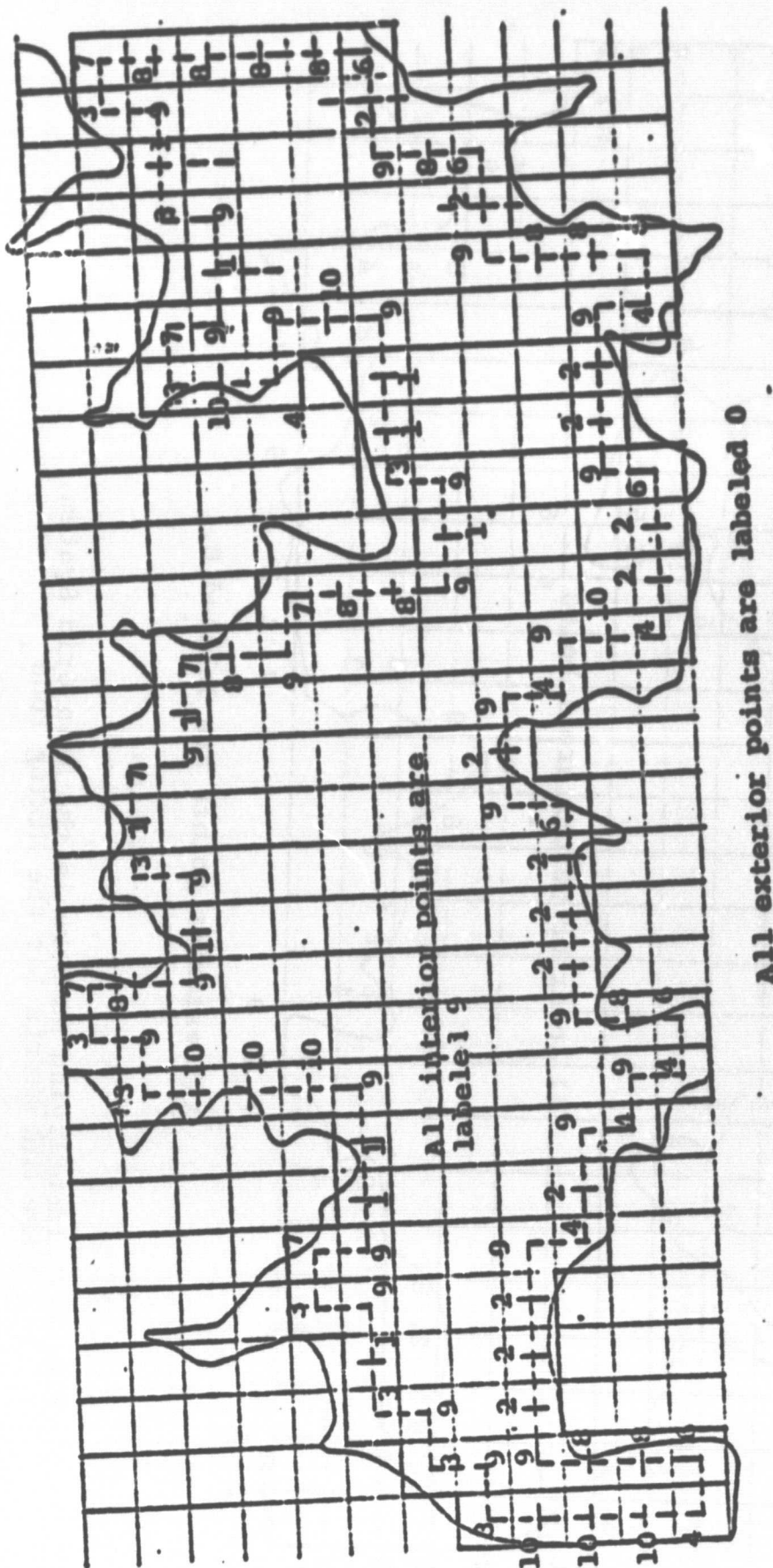


Fig. 7.4 Identifying numbers in the Half Grid System  
The MHI matrix for the Mixing Pond

- (a)  $\tau = 3.2 \times 10^{-6} v^2$ , Elman (1905)  
 (b)  $\tau = 9.3 \times 10^{-6} v^{1.8}$ , Hallstrom (1941)  
 (d)  $\tau = 1.21 \times 10^{-6} v^2 + 2.25 \times 10^{-6} (v-5.6)^2$ ,  
 Van Dorn (1953)  
 (e)  $\tau = 1.98 \times 10^{-6} v^2$ , Wilson (1960)  
 (f)  $\tau = 0.79 \times 10^{-6} v^2$ , Wilson/2.5

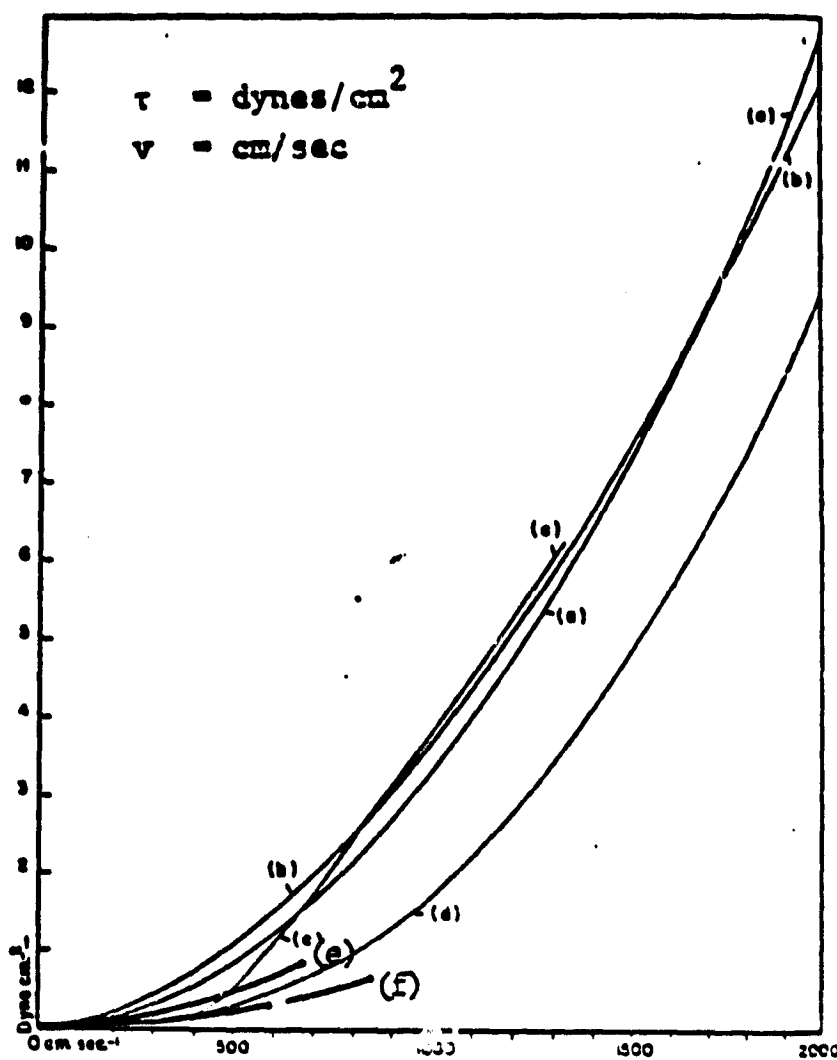


FIGURE 77. Relation of wind stress  $\tau$  to wind velocity  $W$ , based on (a) equation 50,  
 (b) equation 51, (c) Saur's computations after Hunk and Anderson, (d) equation 56.

Fig. 7.5 Comparison of the used wind speed versus  
 surface shear stress relationship with the  
 various suggested relationships.  
 (Reproduction from "A treatise on Limnology"  
 by George Evelyn Hutchinson, 1957)

- ▲ A = Main Lake, horizontal
- ▲ B = Mixing Pond, horizontal
- ▲ C = Vertical, Main Lake and Mixing Pond

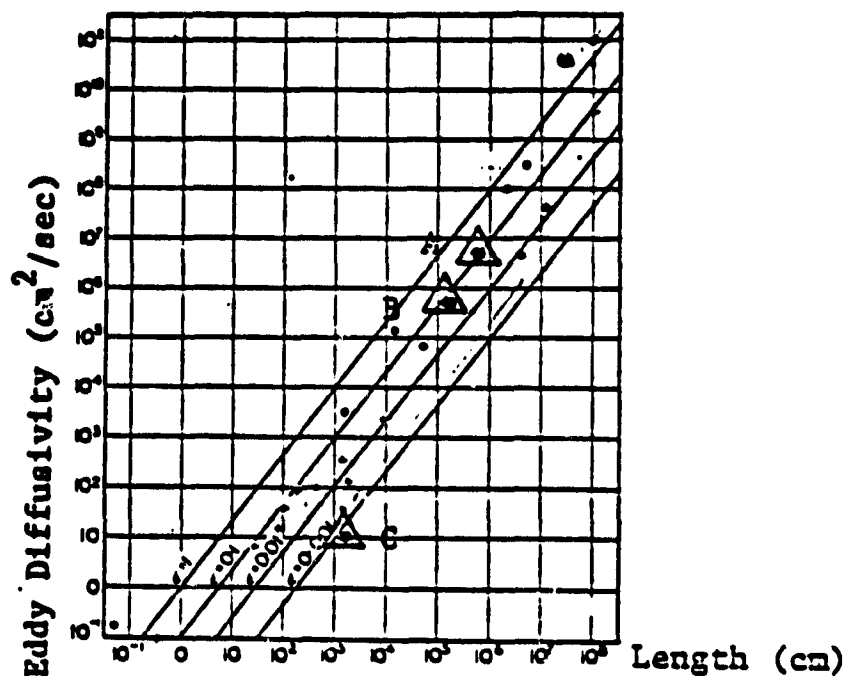


FIG. 167a. The relation  $F(\delta) = \epsilon^{1/3}$  according to observations (logarithmic scale): points, values of Richardson from the atmosphere; crosses, values of Stommel (Blaimore, Bermuda and Woods Hole); triangles, values of Hanzawa.

Fig. 7.6 Comparison of the used turbulent eddy diffusivities with the observed values, (Reproduction from "Physical Oceanography" by Albert Defant, 1961)

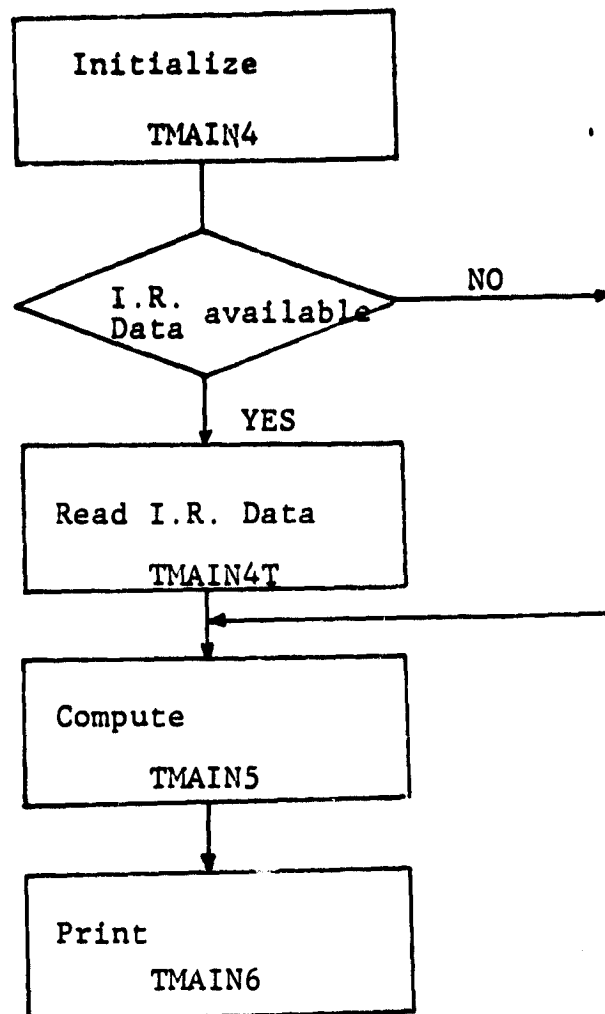


Fig. 7.6a Flow Chart for Program Execution

SOME REPRESENTATIVE RESULTS FOR THE FAR FIELD MODEL  
(UNSTRATIFIED CASE)

Time Step : 6 sec  
 Total Time : 10 Mn  
 Wind : 4.96 m/sec (11.1 mph) 220°  
 Plant Discharge :  $229 \times 10^6$  kg/hr  
 (1012 X 10<sup>6</sup> GPM)  
 Horizontal Viscosity: 45000 cm<sup>2</sup>/sec  
 Vertical Viscosity : 10 cm<sup>2</sup>/sec  
 Depth : Variable

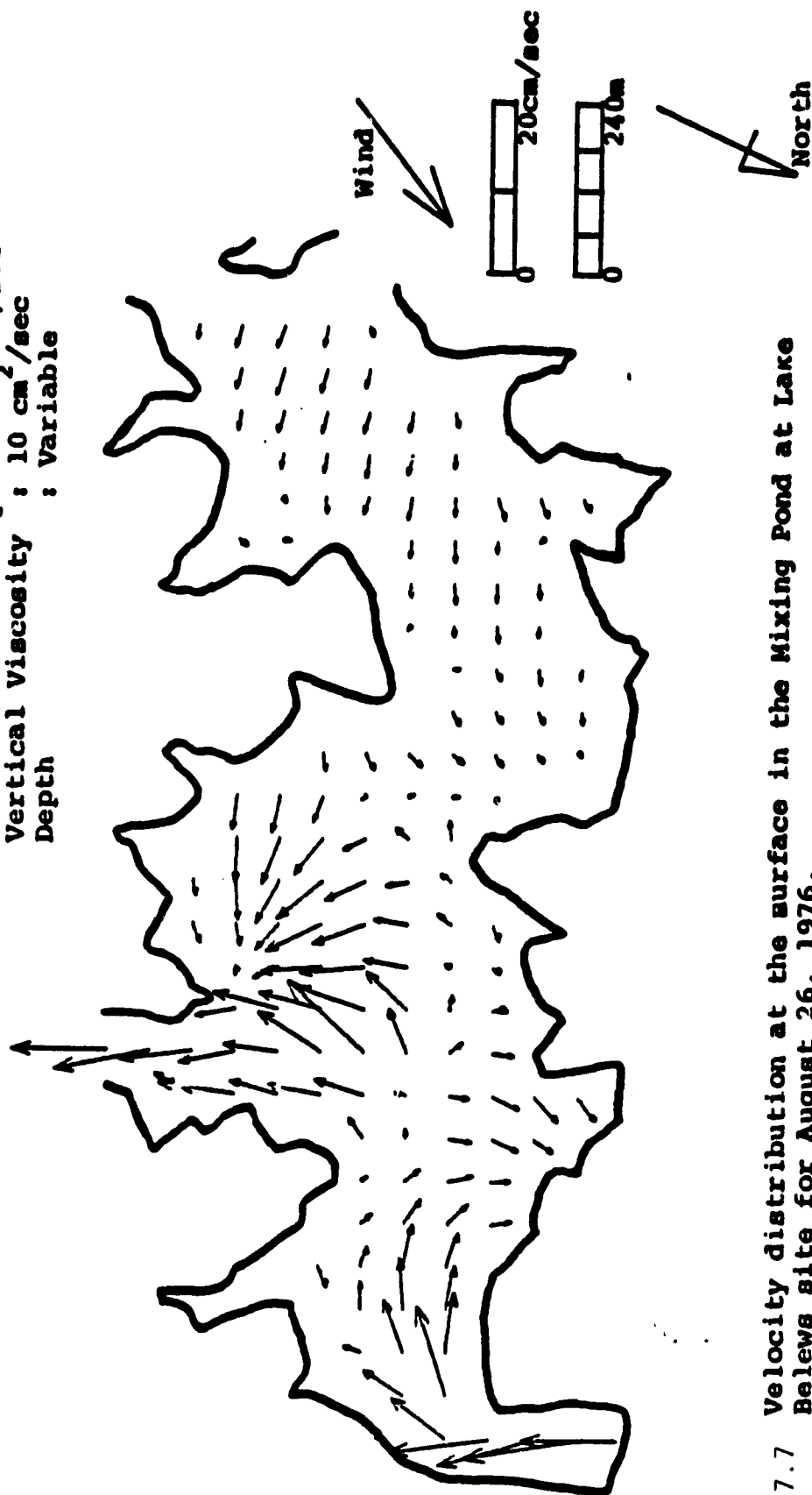


Fig. 7.7 Velocity distribution at the surface in the Mixing Pond at Lake Belews site for August 26, 1976.

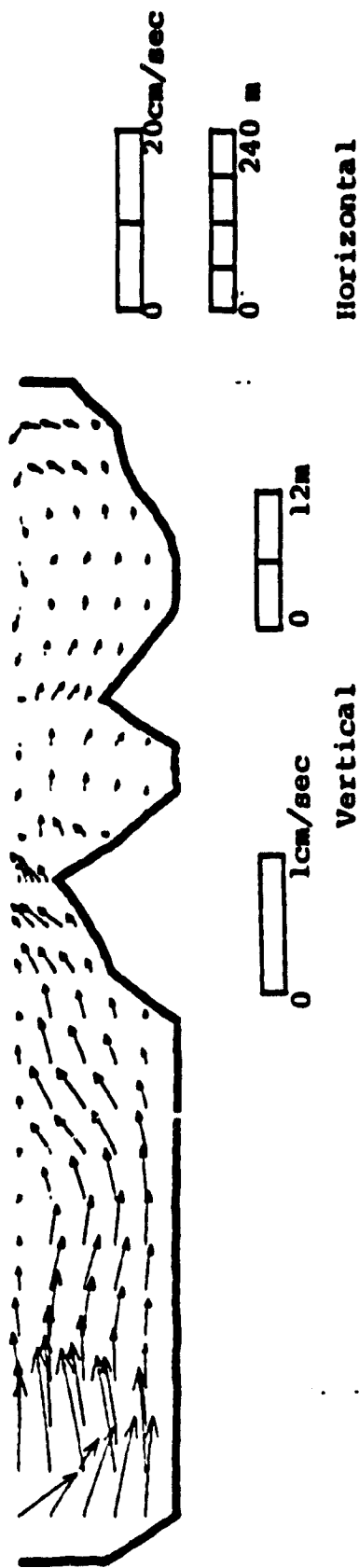


Fig. 7.8 Velocity distribution at vertical section along J=5 in the Mixing Pond at Lake Belevs site for August 26, 1976.

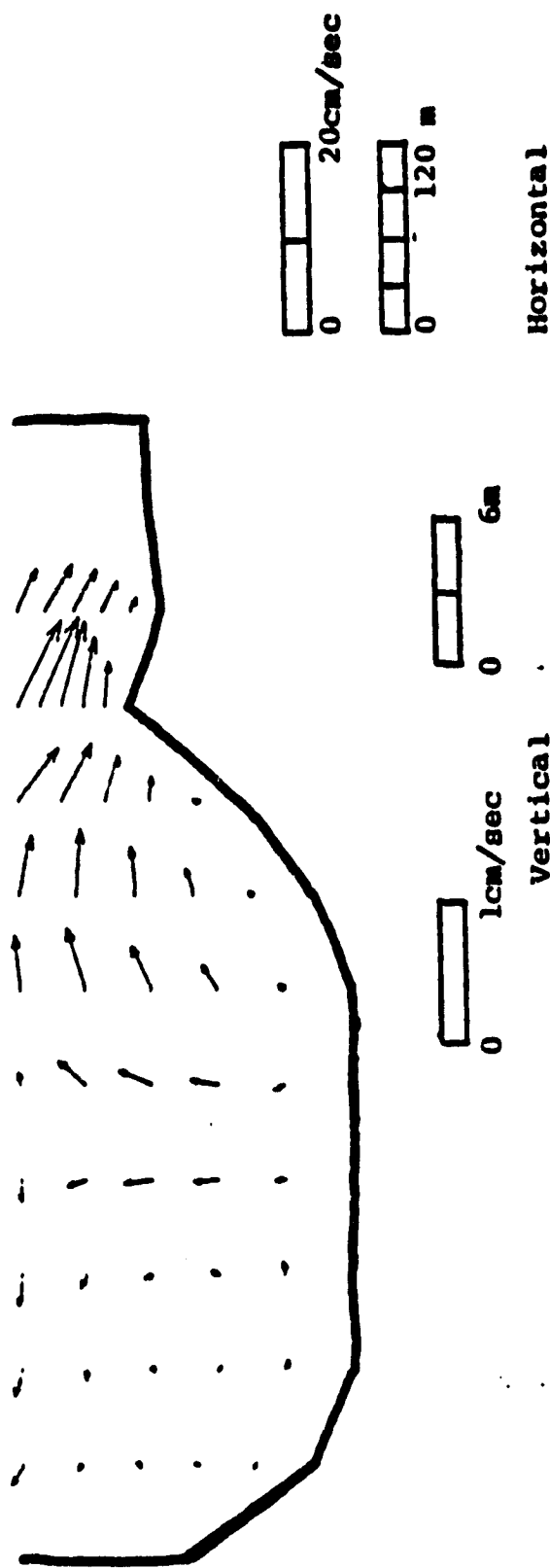


Fig. 7.9 Velocity distribution at vertical section along I=10 in the Mixing Pond at Lake Belevs site for August 26, 1976.

Discharge Temp: 38.9 - 39.1°C  
 Air Temp: 23.9 - 28.9°C  
 Wind: 3.7 - 5.0 m/sec  
 (8.3 - 11.2 mph) SSW

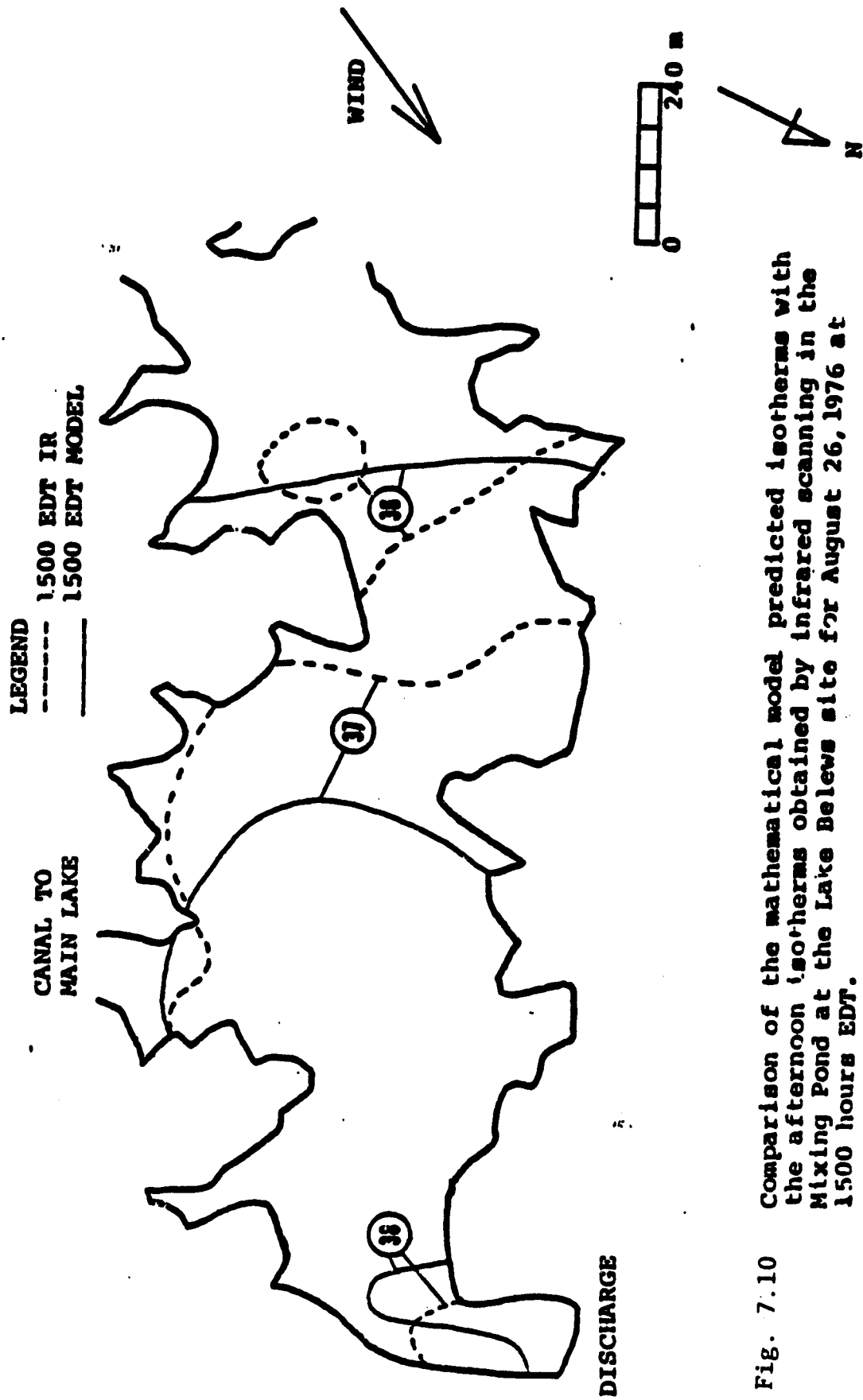


Fig. 7.10 Comparison of the mathematical model predicted isotherms with the afternoon isotherms obtained by infrared scanning in the Mixing Pond at the Lake Belevs site for August 26, 1976 at 1500 hours EDT.



## 8.0 Sample Case, Far-Field (Stratified Case, Lake)

The sample case for far-field stratified and unstratified cases are very much similar. In this section the extra information that is needed for the stratified case will be shown.

8.1 Problem Statement: This is the same as that for an unstratified pond case, except here it is required to find velocity and temperature distributions in a stratified lake.

8.2 Choice of Programs: Since this is a stratified case, the programs to be used are (1) TMAIN 4B, (2) TMAIN5B, TMAIN5TB, TMAIN5VB and (3) TMAIN 6B. The data elements that go with these main programs are (1) DATAML (2) DAML5 and (3) DATAML6. If the initial conditions are known (ie ground truth data and IR data), then the program TMAIN4TB can be used. The data element that goes with it is ITLK1. The authors in this case had initial conditions for August 26, 1977 (Mathavan, 1977) and, therefore, they used TMAIN 4TB and ITLK1. If the initial conditions are not known, zero velocity and ambient temperature can be used as initial conditions. In this case, the program TMAIN 4TB and the data element ITLK1 that goes with it can be thrown away and yet velocity and temperature distributions can be obtained. The flow chart is shown in Figs (9.2 to 9.4)

8.3 Calculation of Input Parameters: In this section, the specification of the grid system, and the reference quantities chosen, will be presented first followed by the actual calculation of input parameters. Only those quantities that did

not appear in the unstratified section and are necessary for stratified case, will be presented.

(1) Grid System: The computational domain of 2880 meters by 6720 meters in the horizontal plane as shown in Fig. (8.1) covers most of the lake. A grid size of 240 meters by 240 meters gives a 23 X 13 mesh in the horizontal plane. The depth is divided into five layers, thus giving a 29 X 13 X 6 mesh for the entire computational domain.

(2) Reference Quantities: Reference quantities are used to non-dimensionalize the input parameters and they are given below:

Reference velocity : 30 cm/sec

Reference temperature: 30°C

Reference horizontal length: 6732 meters

Reference depth = Maximum depth = 40 meters

Reference Horizontal eddy viscosity =  $4.5 \times 10^6 \text{ cm}^2/\text{sec}$

Reference vertical eddy viscosity =  $10 \text{ cm}^2/\text{sec}$

$Pr = 1$  is chosen

Reference horizontal eddy diffusivity =  $4.5 \times 10^6 \text{ cm}^2/\text{sec}$

Reference vertical eddy diffusivity =  $10 \text{ cm}^2/\text{sec}$

### 8.3.1 Input Data for Data Element "DATAML" (For main program TMAIN 4B)

The calculation of the first 12 input data cards is similar to "INDATA" which was explained in section 7.3 of the previous section. Here the rest of the input data will be explained.

<u>CARD NO.</u>	<u>FORTRAN QUANTITY</u>
28	MAR(1,1),MAR(2,1),MAR(3,1)
35	-----

This is to be selected based on the domain type and an example how to make the choice is given in Fig.(7.4). This data can be seen in the input data in section 3.4).

<u>CARD NO.</u>	<u>FORTRAN QUANTITY</u>
36	MRH(1,1), MRH(2,1)
52	-----

The procedure for determining MRH is similar to MAR and can be determined very easily as is explained in Fig. (7.5) This data can be seen in the input data in section (8.4).

CONTINUED

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
53	HI(1,1), HI(1,2)...
.	HI(2,1), HI(2,2)...
.	-----
.	-----
81	-----

This set of input data cards represents the non-dimensional depth matrix. The depth data for Lake Belews site is obtained from geological survey maps and they are non-dimensionalized using a reference depth of 40 meters. This data can be seen in the input data in section (8.4)

### 8.3.2 Calculation of Input Data for Data Element "DATAML5"

(For main programs TMAIN5B, TMAIN5TB and TMAIN5VB)

The first 13 cards are the same as that of "DATAML" which are explained in the section 3.3.1. Now the rest of the input data will be explained.

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
14	NIN,NOUT

Number of Inlet nodes (NIN) in this case = 3

Number outlet nodes (NOUT) in this case = 3

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
15	I,J,K V(I,J,K), V(I,J,K), T(I,J,K)
16	
17	-----

This set of input data cards are for the inlet into the lake. The discharge is introduced at three grid points identified by I=27, J=3 and K= 1,2 and 3. The discharge velocity is calculated using the following quantities.

Plant discharge =  $228.9 \times 10^6 \text{ kg}$

Discharge depth = 8 meters

Discharge width = 240 meters

The discharge is in the x-direction (ie v-velocity only, u-velocity is zero)

The v-velocity is calculated as follows:

$$V_i = \frac{228 \times 10^6 \times 10^3}{.2 \times 3600 \times 2400 \times 800 \times 0.9926}$$

$$= 3.336 \text{ cm/sec}$$

This velocity is non-dimensionalized with reference velocity which is equal to 30 cm/sec.

$$v(I,J,K) = \frac{3.336}{30} = 0.111$$

The discharge temperature =  $39.0^\circ\text{C}$

$$\text{Non-dimensional temperature} = \frac{39 - 30}{30} = 0.1833$$

$$T(I,J,K) = 0.1833$$

<u>CARD NO.</u>	<u>FORTTRAN QUANTITY</u>
18	I,J,K U(I,J,K), V(I,J,K)
19	-----
20	-----

This set of cards are for outlet velocity and outlet velocity is in the y-direction. The outlet velocity is equal to the inlet velocity.

$$U_o = 3.336 \text{ cm/sec}$$

which after non-dimensionalization gives

$$U(I,J,K) = \frac{U_o}{U_{ref}} = \frac{3.336}{30} = 0.111$$

$$V(I,J,K) = U(I,J,K) \left[ \text{ie, } V_1(27,3,K) = U_o(19,2,K) \right]$$

the outlet velocity is at  $I = 19$ ,  $J=2$ ,  $K = 1,2$  and  $3$ .

### 8.3.3 Calculation of Input Data for "DATAML6" (for main programs TMAIN6)

Same as first 13 lines of "DATAML5"

### 8.4 Sample Input (For Far Field Stratified Case)

In the previous section the calculation of Input parameters for different programs is presented. In this section the calculated numerical values are summarized in order for each main program.



#### 8.4.1 Sample Input for Main Program TMAIN4B (DATAML)

[illegible]

OF POOR QUALITY

5,0,0,0,0,0,0,0,0,0,0,0,0,42,0,40,0,95,0,72,0,46,0,50  
6,0,0,0,0,0,0,0,0,0,0,0,0,36,0,72,0,95,0,53,0,27,0,  
7,0,0,0,0,0,0,0,0,0,0,0,0,26,0,32,0,65,0,95,0,42,0,21,0,  
8,0,0,0,0,0,0,0,0,0,0,0,0,19,0,34,0,42,0,42,0,30,0,19,0,19  
9,0,0,0,0,0,0,0,0,28,0,28,0,27,0,34,0,42,0,42,0,30,0,27,0,19  
10,0,0,0,0,19,0,19,0,28,0,57,0,76,0,80,0,88,0,50,0,30,0,27,0,19  
11,0,0,0,0,19,0,27,0,50,0,65,0,46,0,46,0,88,0,50,0,30,0,19,0,19  
12,0,0,0,19,0,19,0,27,0,57,0,80,0,31,0,34,0,91,0,50,0,24,0,19,0,  
13,0,0,0,19,0,34,0,69,0,80,0,40,0,31,0,46,0,91,0,50,0,24,0,0,  
14,0,0,0,30,0,61,0,50,0,57,0,28,0,33,0,37,0,91,0,50,0,24,0,0,  
15,0,24,0,24,0,50,0,54,0,23,0,28,0,26,0,57,0,88,0,50,0,24,0,0,  
16,0,24,0,50,0,50,0,19,0,21,0,0,28,0,57,0,80,0,50,0,24,0,0,  
17,0,32,0,32,0,32,0,26,0,0,0,0,40,0,80,0,40,0,24,0,24,0,0,  
18,0,32,0,32,0,32,0,0,0,0,0,36,0,72,0,36,0,0,0,0,0,  
19,0,32,0,32,0,32,0,0,0,0,0,28,0,57,0,23,0,0,0,0,0,  
20,0,0,0,0,0,0,0,0,0,0,0,0,28,0,57,0,23,0,0,0,0,0,  
21,0,0,0,0,0,0,0,0,0,0,0,0,40,0,80,0,40,0,0,0,0,0,  
22,0,0,0,0,0,0,0,0,0,0,0,0,34,0,69,0,34,0,0,0,0,0,  
23,0,0,0,0,0,0,0,0,32,0,32,0,35,0,76,0,38,0,0,0,0,0,  
24,0,0,0,0,0,0,0,0,32,0,32,0,65,0,65,0,69,0,34,0,0,0,0,0,  
25,0,0,0,0,0,0,0,0,32,0,65,0,65,0,32,0,34,0,34,0,0,0,0,0,  
26,0,0,0,0,0,32,0,32,0,65,0,32,0,32,0,0,0,0,0,0,0,0,  
27,0,0,0,0,32,0,65,0,30,0,19,0,0,0,0,0,0,0,0,0,  
28,0,0,0,0,32,0,65,0,30,0,19,0,0,0,0,0,0,0,0,0,  
29,0,0,0,0,32,0,32,0,57,0,19,0,0,0,0,0,0,0,0,0,

8.4.2 Sample Input for Main Programs TMAIN5B, TMAIN5TB  
and TMAIN5VB (DATA:IL5)

```

1      20      20
2      0.00000000,1.9810
3      1.0,0.000000,0.3.114,1.0
4      0.01,100,1.8,1.0
5      0.03565,0.03565,0.2
6      0.0
7      0.00000000
8      1.0,0.000000,0.3.114
9      1.000000,-0.000000,-0.000000
10     10.0
11     4360.0,0.0,0.0
12     31.7,0.698,1.37,-1.32
13     250.0,0.000000,0.000000
14     3.3
15     27,3,1,0.0,0.111,0.1333
16     27,3,2,0.0,0.111,0.1813
17     27,3,3,0.0,0.111,0.1813
18     19,2,1,0.111,0.0
19     19,2,2,0.111,0.0
20     19,2,3,0.111,0.0

```

## 3.4.3 Sample Input for Main Program TMAIN6 (DATA1L6)

<u>CARD NO.</u>	<u>FORMAT &amp; VALUE</u>
1	2C 13
2	0.0002222,1.9309
3	1.0,0.002228,63.114,1.0
4	0.01,100,1.3,1.0
5	0.03565,0.03565,0.2
6	0.0
7	0.0002673
8	1.0,0.002228,63.114
9	1.000428,-0.000019,-0.0000046
10	33.0
11	4360.0,0.0,0.0
12	31.7,0.695,1.37,-1.32
13	0.0,0.000667,0.0000667

### 8.5 Program Execution Procedure

The execution procedure for far field stratified case is similar to the unstratified case which is explained in section (7.5), except TMAIN 4T, TMAIN 5 and TMAIN 6 have to be replaced by TMAIN 5TB, TMAIN 5B and TMAIN 6B and the data elements ITPK1, INDATA 5 have to be changed to ITLK1, DATAIL5.

### 8.6 Sample Output

The sample output for far field stratified is similar to the near field which is given in section (6.6).

SOME REPRESENTATIVE RESULTS FOR  
FAR-FIELD MODEL (STRATIFIED CASE)



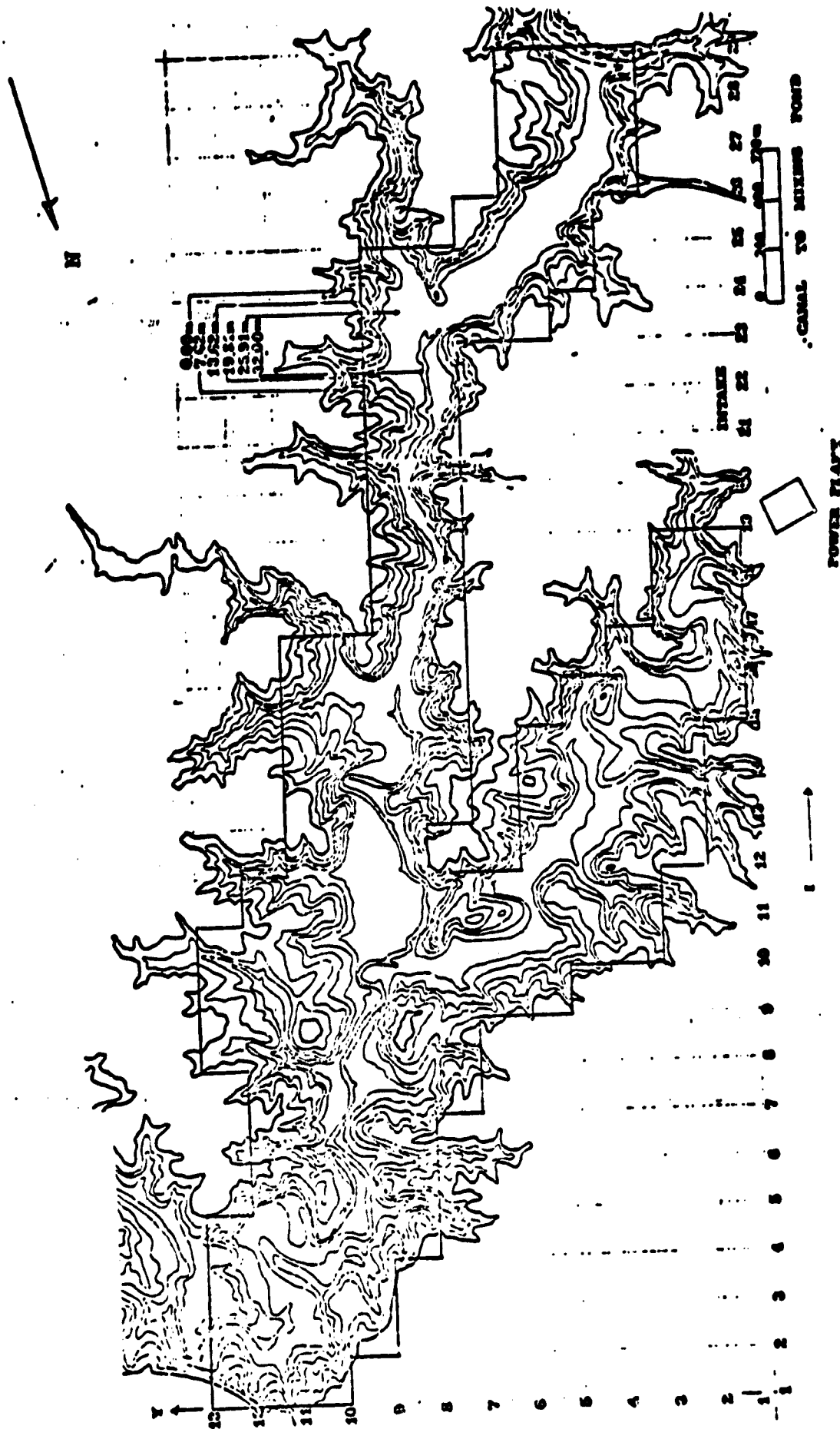


Fig. 8.1 COMPUTATIONAL GRID FOR MAIN LAKE AT BELTUS LAKE SITE

Time Step : 1 sec  
 Total Time : 75 sec  
 Wind : 4.96m/sec (11.1 mph) 220°  
 Plant Discharge : 229 X 10<sup>6</sup> kg/hr  
 (1012 X 10<sup>3</sup> GPM)  
 Horizontal Viscosity: 4.5 X 10<sup>6</sup> cm<sup>2</sup>/sec  
 Vertical Viscosity : 10-0.1 cm<sup>2</sup>/sec  
 Depth : Variable

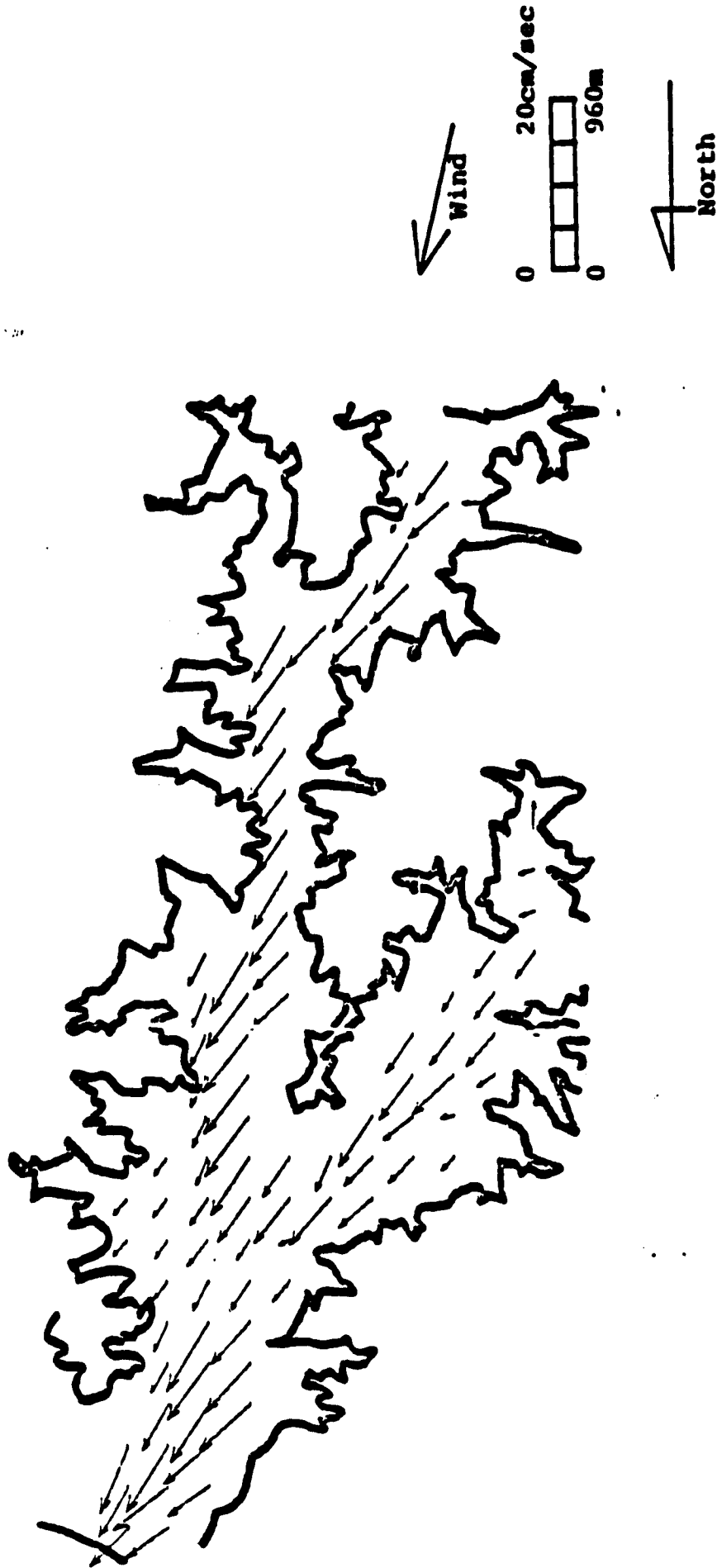


Fig. 8. Velocity distribution at the surface in the Main Lake at Lake Belevs site for August 26, 1976.

Time Step : 1 sec  
 Total Time : 75 sec  
 Wind : 4.96 m/sec (11.1 mph) 220°  
 Plant Discharge : 229 X 10<sup>6</sup> kg/hr  
 (1012 X 10<sup>6</sup> cm<sup>2</sup>/sec)  
 Horizontal Viscosity: 4.5 X 10<sup>6</sup> cm<sup>2</sup>/sec  
 Vertical Viscosity : 10-0.1 cm<sup>2</sup>/sec  
 Depth : Variable

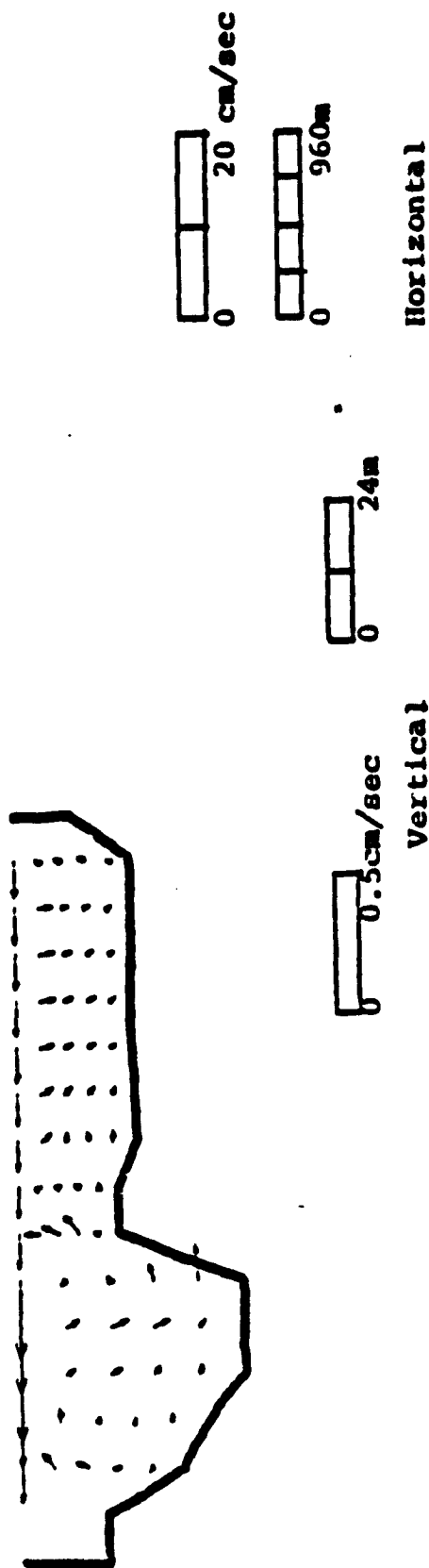


Fig. 8.3 Velocity distribution at vertical section along J=10 in the Main Lake at Lake Belevs site for August 26, 1976.

Time Step : 1 sec  
 Total Time: 75 sec  
 Wind : 4.96 m/sec (11.1 mph) 220°  
 Plant Discharge :  $229 \times 10^6$  kg/hr  
 (1012  $\times 10^3$  GPM)  
 Horizontal Viscosity:  $4.5 \times 10^6$  cm<sup>2</sup>/sec  
 Vertical Viscosity: 10-0.1 cm<sup>2</sup>/sec  
 Depth : Variable

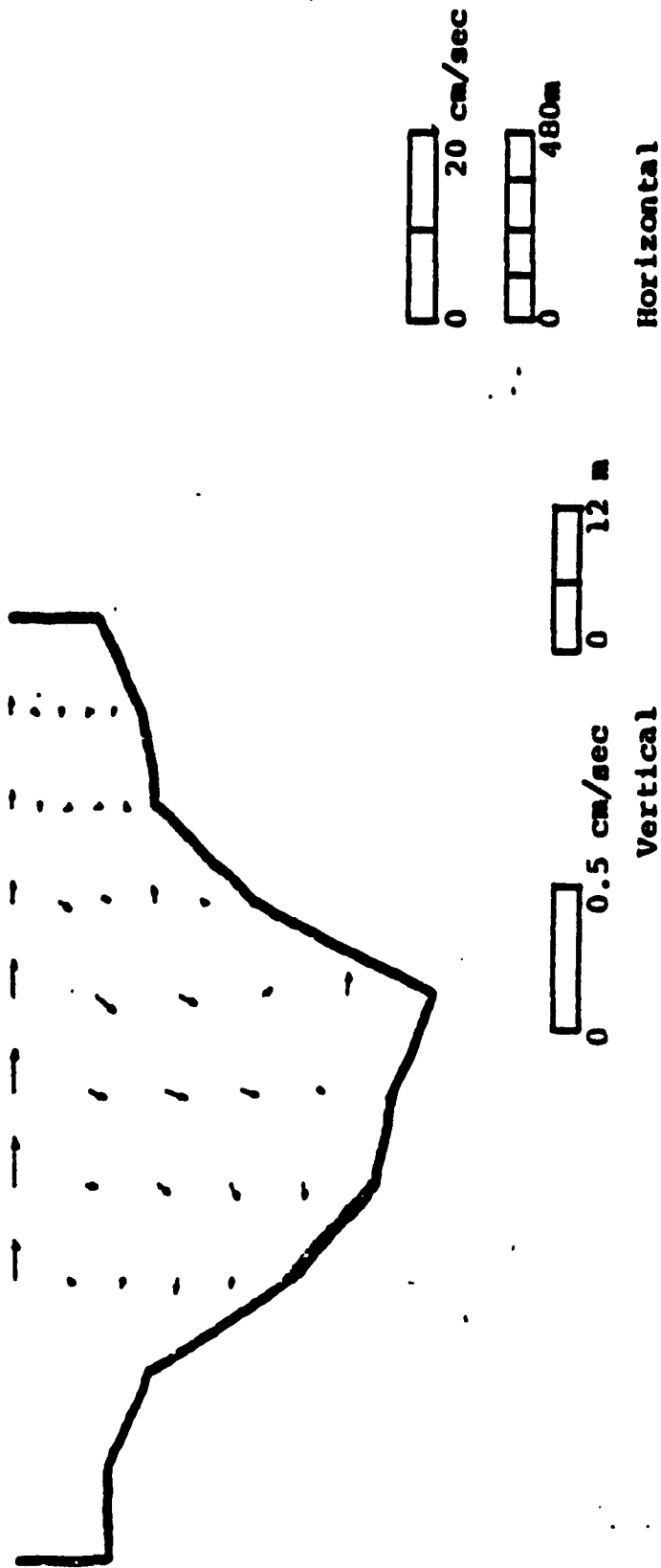


Fig. 8.9 Velocity distribution at vertical section along I=10 in the Main Lake at Lake Belevs site for August 26, 1976.

Discharge Temp: 38.9 - 39.1°C  
 Air Temp : 23.9 - 28.9°C  
 Wind : 3.7 - 5.0 m/sec  
 (8.3 - 11.2 mph) S-SW

# LEGEND

— 1500 EDT IR  
 --- 1500 EDT MODEL

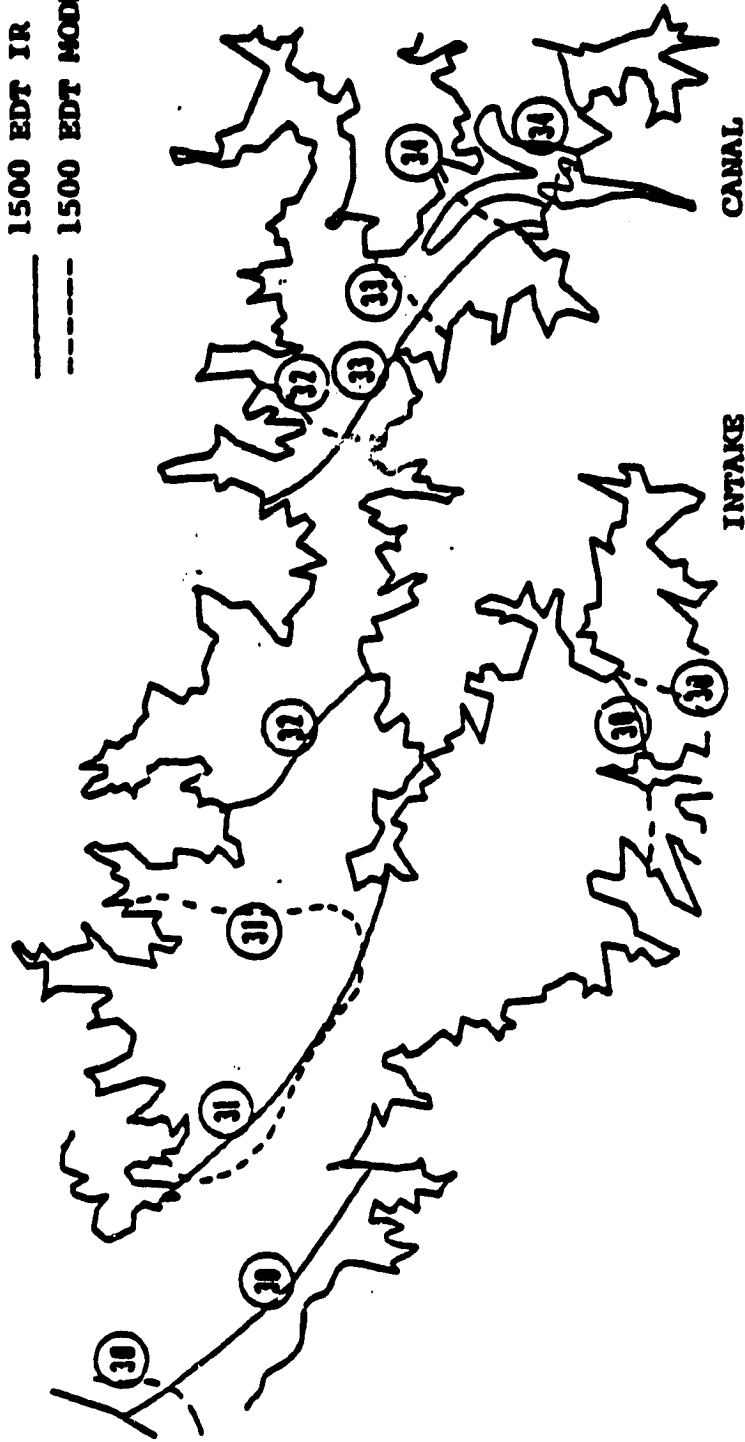


Fig. 8.10 Comparison of the mathematical predicted isotherms with the afternoon isotherms obtained by infrared scanning in the Main Lake at Lake Belevs site for August 26, 1976 at 1500 hours EDT.

## REFERENCES

1. Wilson, B. W., "Note on Surface Wind Stress Over Water at Low and High Wind Speeds, "Journal of Geophysical Research", Vol. 65, No. 10, 1960.
2. Venkata, J., "A Numerical Investigation of Thermal Plumes", Ph.D. thesis submitted to the University of Miami, August, 1977.
3. Mathavan, S.K., "Experimental and Numerical Study of Current and Temperature Fields in Lake Belows. An Artificial Cooling Pond", Ph.D. thesis submitted to the University of Miami, August, 1977.
4. Lee, S.S. and Sengupta, S., "Three Dimensional Thermal Pollution Models", Phase III, prepared for National Aeronautics and Space Administration under contract NASA 10-8926, December, 1976.
5. Harleman, D.R.F., and Stolzenback, K.D., "Engineering and Environmental Aspects of Heat Disposal from Power Generation." Dept. of Civil Engineering, M.I.T., June, 1975.
6. Parker, F.L. and Krenkel, P.A., Engineering Aspects of Thermal Pollution (1968).
7. Sengupta, S., and Lick, W., "A Numerical Model for Wind Driven Circulation and Temperature Field in Lakes and Ponds", FTAS/TR-74-99, Case Western Reserve University (1974)
8. Documentation of 3 Dimensional Mathematical Models for Thermal Pollution Studies. Vols. I, and III.

# APPENDIX A THE EQUILIBRIUM TEMPERATURE AND THE SURFACE HEAT TRANSFER COEFFICIENT

The net heat transfer through a water surface is composed of radiation penetrating the water surface from above, radiation out of the water surface, evaporation, and conduction transfer. These are indicated schematically in the following figure.

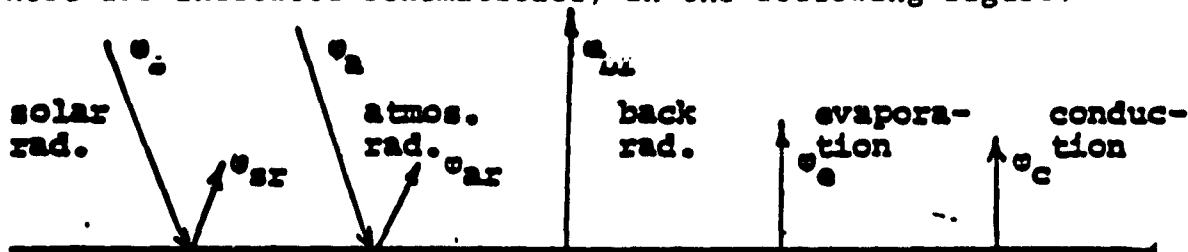


Fig. 1 Heat Transfer Mechanisms at the Water Surface

The following heat balance results,

$$\phi_n = \underbrace{\phi_{sr}}_{\phi_{sn}} + \underbrace{\phi_{ar}}_{\phi_{an}} - \phi_{br} - \phi_e - \phi_c \dots\dots\dots (A-1a)$$

where  $\phi_n$  = net heat input =  $\phi_{sn} + \phi_{an} - \phi_{br} - \phi_e - \phi_c \dots\dots (A-1b)$

Now, equation (A-1) may be rewritten as,

$$\phi_n = \phi_r - \phi_L \dots\dots\dots (A-2)$$

Where,  $\phi_r$  = net absorbed radiation =  $\phi_{sn} + \phi_{an}$

and  $\phi_L = \phi_{br} + \phi_e + \phi_c$

## A.1 Equilibrium Temperature Calculation, $T_e$ (See Appendix<sup>B</sup>)

Under equilibrium conditions equation (A-2) yields,

$$\phi_n = 0 = \phi_r - \phi_L$$

so that

$$\phi_r = \phi_L \dots\dots\dots (A-3)$$



Then by using the approximate formulae in Hareman et al (1975) we obtain by setting  $T_s = T_a$ ,

$$\begin{aligned} & 0.94 \varphi_{sc} (1 - 0.65C^2) + 1.16 \times 10^{-13} (T_a^*)^6 (1 + 0.17C^2) \\ & = 0.97 \varphi (T_a^*)^4 + F(W) [(e_s - e_a) + C_b (T_e - T_a)] \dots (A-4) \end{aligned}$$

Where  $\varphi_{sc}$  = clear sky solar radiation

$C$  = cloudiness ratio

$T_a$  = air temperature ( $^{\circ}\text{C}$  or  $^{\circ}\text{F}$ )

$T_e$  = equilibrium temperature ( $^{\circ}\text{C}$  or  $^{\circ}\text{F}$ )

$T^*$  = absolute temperature ( $^{\circ}\text{K}$  or  $^{\circ}\text{R}$ )

$F(W)$  = windspeed function ( $\text{BTU}/\text{ft}^2/\text{day}$ , mm Hg)

$e_s$  = saturated vapor pressure at water surface temperature (mm Hg)

$e_a$  = saturated vapor pressure at air temperature (mm Hg)

$\sigma$  = Stefan-Boltzmann constant  $\approx 4.1 \times 10^{-8} \text{ BTU}/\text{ft}^2/\text{day}, ^{\circ}\text{R}^4$

$C_b$  = Bowen constant =  $0.255 \text{ mm Hg}/^{\circ}\text{F}$  (see Appendix B)

$W$  = windspeed (mph)

For natural water surface,

$$F(W) = 17W \dots \dots \dots (A-5a)$$

and, for an artificially heated surface,

$$F(W) = 22.4 (T_e - T_a)^{1/3} + 17W \dots \dots \dots (A-5b)$$

Thus, equation (A-4) becomes,

$$\begin{aligned} & 0.94 \varphi_{sc} (1 - 0.65C^2) + 1.16 \times 10^{-13} (T_a^*)^6 (1 + 0.17C^2) \\ & = 0.97 \varphi (T_a^*)^4 + 17W [(e_s - e_a) + 0.255 (T_e - T_a)] \dots (A-6) \end{aligned}$$

Location - Miami (latitude  $26^{\circ}\text{N}$ )

Date - December 20

From CRC (1970),

$$\text{at } T_a = 25^{\circ}\text{C} \rightarrow e_a \approx 0.43 \text{ psia}$$

$$\text{at } T_a = 27^{\circ}\text{C} \text{ as guess } \approx e_s = e_e \approx 0.51 \text{ psia}$$

From Harleman et al (1975),

$$\begin{aligned} \phi_{sc} &\approx 425 \text{ Langleys/day} = 1560 \text{ BTU/ft}^2/\text{day} \dots \text{ using} \\ &100\% \text{ sunshine curve at } 26^{\circ}\text{N, Dec. 20} \end{aligned}$$

Note: 1 Langley/min. =  $220.62 \text{ BTU/ft}^2/\text{day}$  ... using

100% sunshine curve at  $26^{\circ}\text{N}$ , December 20

Note: 1 Langley/min. =  $220.62 \text{ BTU/ft}^2$ , hr. =  $1 \text{ calorie/cm}^2/\text{min.}$

Then using equation (C-6) with  $C=0$

$$\begin{aligned} 0.94 (1560) (1) + 1.16 \times 10^{-13} (5.37 \times 10^2)^6 (1) &= 4250 \\ 4 \times 10^{-8} (5.406 \times 10^2)^4 + 170 [ (.255) (2) + (.08) (51.7) ] \\ 4206 \text{ close enough!} \end{aligned}$$

$$\therefore T_e \approx 27^{\circ}\text{C}$$

(where 1 psia = 51.7 mm Hg)

Then from equation (A-8),

$$\begin{aligned} K &= 3.88 \times 4.1 \times 10^{-8} (5.406 \times 10^2)^3 + 170 (.255 + 0.0251 \times 51.7) \\ K &\approx 290 \text{ BTU/ft}^2, ^{\circ}\text{F, day} \end{aligned}$$

$$\text{where } \left. \frac{\partial e_s}{\partial T} \right|_{T=T_{av}} \approx \frac{e_s - e_a}{T_e - T_a} = .0251$$

or,

$$\begin{aligned}
 & 0.94 \varphi_{sc} (1 - 0.65C^2) + 1.16 \times 10^{-13} (T_a^*)^6 (1 + 0.17C^2) \\
 & = 0.97\varphi (T_a^*)^4 + [22.4 (T_e - T_a)^{1/3} + 14W] \\
 & \quad \cdot [(e_s - e_a) + 0.255 (T_e - T_a)] \text{----- (A-7)}
 \end{aligned}$$

Therefore, for known  $\varphi_{sc}$ ,  $e_s$ ,  $e_a$ ,  $T_a$  and  $W$ ,  $T_e$  can be determined by trial and error methods.

#### A.2 Surface Heat Transfer Coefficient (K)

From Harleman et al (1975) the surface heat transfer coefficient K, can be determined as follows

$$K = \frac{\partial \varphi_L}{\partial T_{av}} = \frac{\partial \varphi_n}{\partial T_{av}}, \text{ since } \varphi_r \neq \varphi_r(T_s) \text{ and } \therefore \frac{\partial \varphi_r}{\partial T_{av}} = 0.$$

where  $T_{av} = (T_s + T_e)/2$

Thus,

$$\begin{aligned}
 K &= 3.88\sigma (T_{av}^*)^3 + F(W) \left[ \left( \frac{\partial e_s}{\partial T} \right)_{T=T_{av}} + C_b \right] \\
 &+ [(e_{av} - e_a) + C_b (T_{av} - T_a)] \frac{\partial F(W)}{\partial T_{av}} \text{----- (A-8)}
 \end{aligned}$$

Where

$$\frac{\partial F(W)}{\partial T_{av}} = \begin{cases} 0 & \text{for natural water surface} \\ 1/3(22.4) (T_{av} - T_a)^{-2/3} & \text{for artificially heated water surface} \end{cases}$$

#### A.3 Numerical Example

Consider natural water surface

$$C = 0$$

$$T_a = 25^\circ\text{C}$$

$$W = 10 \text{ mph}$$

#### A.4 Discussion

The equilibrium surface temperature,  $T_e$ , for a natural water surface, can be greater than the atmospheric temperature,  $T_a$ , whereby  $T_s$  increases from values below  $T_a$  up to  $T_e$  as equilibrium is reached. As can be seen in the figure below  $T_e$  can be greater or smaller than  $T_a$  depending on the time of the day. Simply  $T_e > T_a$  during the hours of sunshine and  $T_e < T_a$  at night when the water surface is cooling.

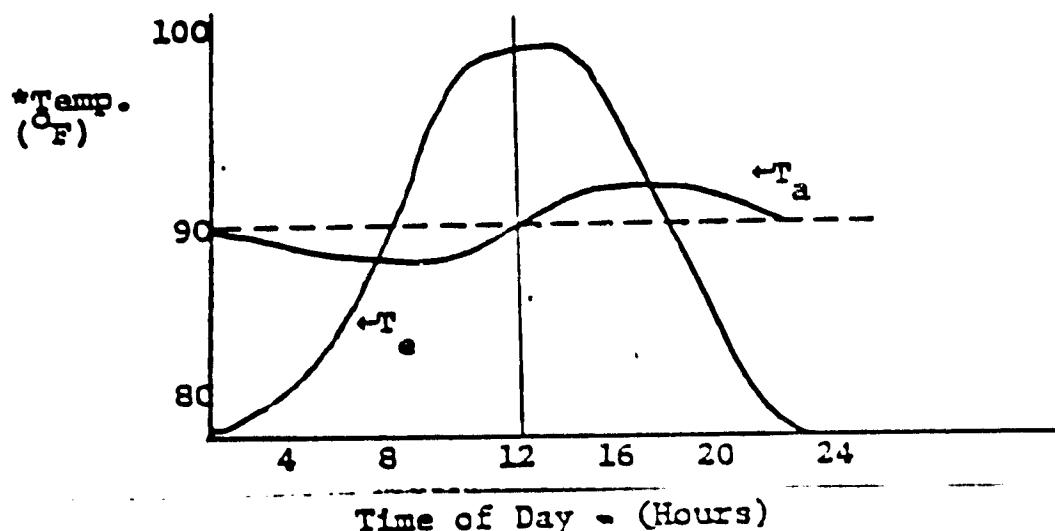


Fig. ( 2 ) Daily variation of air temperature and computed equilibrium temperature.

\* This plot is taken from Parker (1968) and has no relation to the numerical example given in this paper. However, the numerical example considered 100% possible hours of sunshine

## APPENDIX B

## HEAT TRANSFER MECHANISMS

The analysis in this section is taken from Harleman et al. (1975) and is summarized.

B.1 Solar Radiation (short wave)

The incident solar radiation impinging on the water surface may be expressed as

$$\varphi_s = \varphi_{sc}(1-0.65C^2)$$

Where  $\varphi_{sc}$  = clear sky solar radiation obtained using the 100% possible sunshine curve (given in Appendix A)

C = fraction of sky covered by clouds.

The reflected solar radiation is typically 6% of incident solar radiation, hence the net solar radiation absorbed by the water surface is,

$$\varphi_{sn} = \varphi_s - \varphi_{sr} \approx 0.94\varphi_{sc}(1-0.65C^2)$$

B.2 Atmospheric Radiation (long wave)

The basic equation for the incident atmospheric radiation, is given as

$$\varphi_a = \epsilon \sigma T_a^{*4}$$

Where  $\epsilon$  = average emittance of the atmosphere

$\sigma$  = Stefan-Boltzmann constant

$T_a^*$  = air temperature (absolute)

However, good agreement with experimental data has indicated that  $\epsilon$  is a function of  $T_a$ , and specifically,  $T_a^{*b}$  dependence gives best results for atmospheric radiation at low temperatures.

as well as providing a good fit at high temperatures. Clear sky incident atmospheric radiation,  $\varphi_{ac}$ , may be expressed as,

$$\varphi_{ac} = 1.2 \times 10^{-13} (T_a^*)^6$$

and, then incident atmospheric radiation including cloudiness may be expressed as,

$$\varphi_a = \varphi_{ac} (1 + 0.17c^2)$$

A figure of 3% is usually accepted as reflectance of a water surface to longwave radiation. Thus the net atmospheric radiation absorbed by the surface is

$$\varphi_{an} = \varphi_a - \varphi_{ar} = 0.97\varphi_a$$

and, therefore, we have

$$\varphi_{an} = 1.16 \times 10^{-13} (T_a^*)^6 (1 + 0.17c^2)$$

### B.3 Longwave Radiation from the Water Surface, $\varphi_{br}$

Harleman et al (1975) note that the emissivity of a water surface is independent of temperature and salt or colloidal concentrations, and gives a value of 0.97. Thus we obtain,

$$\varphi_{br} = 0.97\sigma(T_s^*)^4$$

Where  $T_s$  = water surface temperature.

### B.4 Evaporative Heat Flux, $\varphi_e$

Evaporation from a water surface occurs as a result of both forced (wind-driven) convection and free (bouyancy driven) convection. The evaporation from a water surface is usually

written (mass/area/time) as

$$E = \rho F(W_z) (e_s - e_z)$$

Where,  $E$  = mass flux (mass/area/time)

$\rho$  = density of water

$W_z$  = windspeed at height  $z$  above surface

$F(W_z)$  = windspeed function for mass flux including both  
free and forced convection effects (length/time/  
pressure)

$e_s$  = saturated vapor pressure at  $T_s$

$e_z$  = vapor pressure at height  $z$  above surface

Then writing the above equation in heat units, the evaporative  
heat flux  $\phi_e$ , is given by,

$$\phi_e = F(W_z) (e_s - e_z)$$

Where  $F(W_z)$  = windspeed function for heat flux (energy/area/  
time/pressure)

Now, dropping the  $z$  subscript (and assuming  $W$  measured "z"  
above the surface ~  $W$  at the surface) we may express  $F(W)$   
for a natural water surface and for an artificially heated  
water surface as

$$F(W) = 17W \dots$$

natural water surface

and

$$F(W) = 22.4 (T_s - T_a)^{1/3} + 14W \dots$$

artificially heated

surface.

### B.5 Conduction Heat Flux, $\phi_c$

Bowen (1926) has suggested that conduction can be directly



related to evaporative fluxes by assuming that eddy diffusivities of heat and mass are identical. Thus,

$$\varphi_c = R_b \varphi_e$$

Where  $R_b = C_b \left| \frac{T_s - T_a}{e_s - e_a} \right|$  = Bowen Ratio

and  $C_b$  = Bowen constant = 0.255 mm Hg/°F

and, therefore the conduction heat flux,  $\varphi_c$ , may be expressed

as,

$$\varphi_c = C_b F (W) (T_s - T_a)$$

COMPUTER PROGRAMS

## 9.0 DESCRIPTION OF COMPUTER PROGRAMS

For the readers convenience, the computer programs and their description are included in this section. There are two sets of main programs, one set for the near field and the second for the far-field. These main programs are described along with the flow charts in subsection (9.1.1) Subsection (9.2) describes subroutines for both the near field and far field.

### 9.1 Main Programs for the Near-Field and Far-Field

There are four main programs for the near-field. They are AMAIN1, AMAIN2, TMAIN1 and TMAIN2. AMAIN programs are for velocity only, and TMAIN programs are for velocity and temperature. AMAIN1 and TMAIN1 are to be used for constant depth and AMAIN2 and TMAIN2 for variable depth conditions.

~~INTENTIONALLY BLANK~~

#### 9.1.1 AMAIN 1 (Main Program for Near-Field)

The main program reads in the data, initializes the necessary quantities and coordinates the subroutines and calculates velocities only for a near field problem. The parameter statement defines the size of the computational domain. The subroutine READ3 reads MAR and MRH matrices. The subroutine 'INITIA' sets the velocities in the receiving basin equal to zero. If there is current, the subroutine 'CURNT' has to be called after the subroutine 'INITIA', which would set the velocities everywhere in the domain equal to the CURNT velocity. Then it follows a set of subroutines to calculate the velocity field for the entire domain for the given discharge and ambient conditions.

\*DOLL(1).AMAIN1

```

1      C      CALCULATES U,V,W, FOR VARIABLE DEPTH LAKE WITH STRETCHS
2      C      ALGORITHM
3      PARAMETER INEQ1,JNEQ1,KNEQ1,INNEQ1,JNNEQ1
4      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IN,JN,KN),
5      CRH(IN,JN,KN),WRH(IN,JN,KN),P(IN,JN),C(IN,JN,KN),E(IN,JN,KN),
6      CHLDT(IN,JN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
7      CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MHR(IN,JN),FHI(IN,JN),
8      COPSX(IN,JN),CPSY(IN,JN)
9      DIMENSION A3(KN)
10     IAMI=IN-1
11     READ 1, IFUN
12     READ 1, LN
13     1      FORMAT (15)
14     READ 2, VVIS,ABP
15     A3(1)=VVIS
16     A3(2)=VVIS
17     A3(3)=VVIS
18     A3(4)=VVIS
19     A3(5)=VVIS
20     READ 2, AI,AH,AV,AP
21     READ 2, EPS,MAXIT,OMEGA,ARBP
22     READ 2, DX,DY,DZ
23     READ 2, AA,BB,CC
24     DL2=(DX*DX+DY*DY)/(DX*DX+DY*DY)
25     2      FORMAT (1)
26     IF(IFUN.GT.0) GO TO 3
27     CALL INITIAT(IN,JN,KN,IN,JN,U,V,W,WH,D,E,P,I,J,K,IN,JN,ABBP)
28     CALL READT(I,J,IN,JN,IN,JN,IN,JN,MAR,MHR)
29     CALL HEIGHT(I,J,K,IN,JN,KN,HI,HX,HY,CC)
30     CALL INLET(I,J,K,IN,JN,KN,C,V,AA,BB)
31     TTOT=0.0
32     GO TO 4
33     3      CONTINUE
34     REWIND 7
35     READ (7) ((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
36     C((V(I,J,K),K=1,KN),J=1,JN),I=1,IN),
37     C((D(I,J,K),K=1,KN),J=1,JN),I=1,IN),
38     C((E(I,J,K),K=1,KN),J=1,JN),I=1,IN),
39     C((WH(I,J,K),K=1,KN),J=1,JN),I=1,IN),
40     C((W(I,J,K),K=1,KN),J=1,JN),I=1,IN),
41     C((CRH(I,J,K),K=1,KN),J=1,JN),I=1,IN),
42     C((WRH(IN,JN,K),K=1,KN),J=1,JN),I=1,IN),
43     C((P(IN,JN),J=1,JN),I=1,IN),
44     C,((HI(I,J),J=1,JN),I=1,IN),
45     C((HX(I,J),J=1,JN),I=1,IN),
46     C((HY(I,J),J=1,JN),I=1,IN),
47     C((MAR(I,J),J=1,JN),I=1,IN),
48     C((MHR(IN,JN),J=1,JN),I=1,IN),
49     CAI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,TTOT
50     REWIND 7
51     CALL INLET(I,J,K,IN,JN,KN,C,V,AA,CC)
52     4      CONTINUE
53     READ 2, DT
54     DO 5 L=1,LN
55     TTOT=TTOT+DT
56     CALL ERROR(IN,JN,IN,JN,DT,WH,WHLDT,FN,ABBP)

```

```

57 CALL WHTOP(IW,JW,IW,JW,KN,WH,K,MRH)
58 CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IW,JW,WH,MRH)
59 CALL INTE(I,J,K,IN,JN,KN,U,V,W,HZ,HX,HY,MRH,XINT,YINT,A3,AI,
60 CAM,AV,TAUX,TAUY,DX,DY,DZ,D,E,DT,CPSX,CPSY,AP)
61 CALL CORINT(I,J,K,IN,JN,KN,ABP,U,V,XINT,YINT,DZ,HZ,MRH)
62 CALL DPSXY(I,J,IN,JN,IW,JW,IW,JW,CPSX,CPSY,P,DX,DY,MRH)
63 CALL FORCE(I,J,IW,JW,XINT,YINT,WHLOT,DX,DY,HZ,HX,HY,MRH,
64 CPSX,CPSY,FH,AP,IN,JN,IW,JW)
65 CALL PREZ(EPS,MAXIT,IN,JN,P,ITN,CPSX,CPSY,FH,DZ,OMEGA,
66 CHRH,I,J,K,IW,JW,DX,DY,EX,IW,JW,ABP)
67 CALL UV(I,J,K,IW,JW,IN,JN,KN,IW,JW,U,V,D,E,H,G,EX,DY,DZ,
68 CW,DT,AI,AP,AH,AV,AJ,HI,HX,HY,P,MRH)
69 CALL UANVC(I,J,K,IN,JN,KN,ABP,DT,U,V,H,G,HI,MRH)
70 CALL UVTOP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MRH)
71 CALL OUTVEL(I,J,K,IN,JN,KN,H,G)
72 CALL OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
73 CALL OLDUV(I,J,K,IN,JN,KN,H,G,U,V)
74 CALL RWH(I,J,K,IW,JW,IN,JN,KN,IW,JW,U,V,WH,HI,DX,DY,DZ,MRH)
75 5 CONTINUE
76 CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IW,JW,WH,MRH)
77 CALL RWRH(I,J,K,IW,JW,IN,JN,KN,IW,JW,U,V,WH,HI,HX,HY,
78 CDX,DY,DZ,MRH,WH)
79 CALL RWR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,HY,DZ,MRH)
80 CALL STORE(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IW,JW,D,E,
81 CHX,HY,HI,MRH,ABP,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,MRH,TTOT)
82 CALL PREPARA(I,AH,AV,AP,DX,DY,DZ,DT,DZ,MAXIT,CPS,OMEGA,
83 CARBP,TAUX,TAUY,TTOT,MRH,WH,IN,JN,IW,JW)
84 CALL PRECOR(IW,IW,WHLOT,JW,JW,IN,JN)
85 CALL PRINTE(I,J,IW,JW,XINT,YINT,IN,JN)
86 CALL PRORPC(IW,JW,IW,JW,FH)
87 CALL PGPSXY(I,J,IN,JN,CPSX,CPSY)
88 CALL PRITEX(ITN,EX)
89 CALL PRPINT(IW,JW,IW,JW,P)
90 CALL PRUV(I,J,K,IN,JN,KN,U,V)
91 CALL PRWH(IW,JW,K,IW,JW,KN,MRH)
92 END

```

### 9.1.2 AMAIN2 (Main Program for Near-Field)

This is same as AMAIN1, except, this main program is to be used when the depth is variable. The subroutine GRADS computes slopes of the bottom in x and y directions respectively.



DJ\_L(1).A4AIN2

```

1      C      CALCULATES U,V,W, FOR VARIABLE DEPTH LAKE WITH STRETCHING
2      C      ALGORITHM
3      C      PARAMETER IN=18, JN=21, KN=5, IWN=17, JWN=20
4      C      DIMENSION U(IN, JN, KN), V(IN, JN, KN), W(IN, JN, KN), WH(IWN, JWN, KN),
5      C      CWR(IN, JN, KN), WRH(IWN, JWN, KN), P(IWN, JWN), D(IN, JN, KN), E(IN, JN, KN),
6      C      CMHLOT(IWN, JWN), XINT(IN, JN), YINT(IN, JN), H(IN, JN, KN), C(IN, JN, KN),
7      C      CHI(IN, JN), HX(IN, JN), HY(IN, JN), MAR(IN, JN), MRH(IWN, JWN), FH(IWN, JWN),
8      C      CDPX(IN, JN), DPSY(IN, JN), JX(JN)
9      C      DIMENSION A3(KN)
10     C      INM1=IN-1
11     C      READ 1, IRUN
12     C      READ 1, LN, LLN
13     1    FORMAT (16I5)
14     C      READ 2, VVIS, ABR
15     C      A3(1)=VVIS
16     C      A3(2)=VVIS
17     C      A3(3)=VVIS
18     C      A3(4)=VVIS
19     C      A3(5)=VVIS
20     C      READ 2, AI, AH, AV, AP
21     C      READ 2, EPS, MAXIT, OMEGA, ARBP
22     C      READ 2, DX, DY, DZ
23     C      READ 2, AA, BB, CC
24     C      READ 2, TAUX, TAUY
25     C      DL2=DX*DX
26     C      READ 2, DT
27     2    FORMAT ( )
28     C      IF (IRUN.LT.0) GO TO 3
29     C      TTOT=0.0
30     C      CALL INITIA(IN, JN, KN, IWN, JWN, U, V, W, WH, D, E, P, I, J, K, IW, JW, ARBP)
31     C      CALL READ3(I, J, IN, JN, IW, JW, IWA, JWA, MAR, MRH)
32     C      CALL HEIGHT(I, J, K, IN, JN, KN, HI, HX, HY, CC, JX)
33     C      CALL GRADSL(IN, JN, KN, IWN, JWN, HI, HX, HY, MAR, MRH, DX, DY)
34     C      GO TO 4
35     3    CONTINUE
36     C      REWIND 7
37     C      READ (7) ((U(I, J, K), K=1, KN), J=1, JN), I=1, IN),
38     C      C((V(I, J, K), K=1, KN), J=1, JN), I=1, IN),
39     C      C((D(I, J, K), K=1, KN), J=1, JN), I=1, IN),
40     C      C((E(I, J, K), K=1, KN), J=1, JN), I=1, IN),
41     C      C((WH(IW, JW, K), K=1, KN), JW=1, JWN), IW=1, IWN),
42     C      C((W(I, J, K), K=1, KN), J=1, JN), I=1, IN),
43     C      C((WR(I, J, K), K=1, KN), J=1, JN), I=1, IN),
44     C      C((WRH(IW, JW, K), K=1, KN), JW=1, JWN), IW=1, IWN),
45     C      C((P(IW, JW), JW=1, JWN), IW=1, IWN)
46     C      C, ((HI(I, J), J=1, JN), I=1, IN),
47     C      C((HX(I, J), J=1, JN), I=1, IN),
48     C      C((HY(I, J), J=1, JN), I=1, IN),
49     C      C((MAR(I, J), J=1, JN), I=1, IN),
50     C      C((MRH(IW, JW), JW=1, JWN), IW=1, IWN),
51     C      CAI, AH, AV, AP, DX, DY, DZ, DT, TAUX, TAUY, TTOT
52     C      REWIND 7
53     4    CONTINUE
54     C      DO 6 LL=1, LLN
55     C      DO 5 L=1, LN
56     C      TTOT=TTOT+DT

```

```

57 CALL INLET (I,J,K,IN,JN,KN,V,G,AA,BB)
58 CALL ERROR(IWN,IW,JW,DT,WH,WHLDT,KN,MRH)
59 CALL WHTOP(IW,JW,IN,JN,KN,WH,K,MRH)
60 CALL WHATIU(I,J,K,IN,JN,KN,IN,JN,WH,WH,MAR)
61 CALL INTE(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3,AI,
62 CAH,AV,TAUX,TAUY,DX,DY,DZ,D,E,DT,DPSX,CPSY,AP)
63 CALL CGRINT(I,J,K,IN,JN,KN,ABR,U,V,XINT,YINT,DZ,HI,MAR)
64 CALL DPSXY(I,J,IN,JN,IN,JN,IN,JN,DPSX,DPSY,P,DX,DY,MAR)
65 CALL FORCE(I,J,IN,JN,XINT,YINT,WHLDT,DX,DY,HI,HX,HY,MRH,
66 CDPSX,DPSY,FH,AP,IN,JN,IN,JN,RINTX,RINTY,U,V,EUL,AER,MAR,KN)
67 CALL PREZEPS,MAXIT,IN,JN,P,ITN,DPSX,CPSY,FH,DL2,CMEGA,
68 CHRH,I,J,K,IN,JN,DX,DY,EX,IN,JN,ABP)
69 CALL UV(I,J,K,IN,JN,IN,JN,KN,IWN,JN,U,V,D,E,H,G,DX,DY,DZ,
70 CW,DT,AI,AP,AH,AV,A3,HI,HX,HY,P,MAR)
71 CALL UANVC(I,J,K,IN,JN,KN,AEP,DT,U,V,H,G,HI,MAR)
72 CALL UVTOP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
73 CALL CUTVEL(I,J,K,IN,JN,KN,H,G)
74 CALL OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
75 CALL OLDUV(I,J,K,IN,JN,KN,H,G,U,V)
76 CALL RWH(I,J,K,IN,JN,IN,JN,KN,IWN,JN,U,V,WH,HI,DX,DY,DZ,MRH)
77 5 CONTINUE
78 CALL WHATIU(I,J,K,IN,JN,KN,IN,JN,WH,WH,MAR)
79 CALL RWRH(I,J,K,IN,JN,IN,JN,KN,IWN,JN,U,V,WH,HI,HX,HY,
80 CDX,DY,DZ,MRH,WRH)
81 CALL RWR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,HY,DZ,MAR)
82 CALL STORE(U,V,WH,P,I,J,K,IN,JW,IN,JN,KN,IN,JN,D,E,
83 CHX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,PH,TTCT)
84 CALL PRPAPA(AI,AH,AV,AP,DX,DY,DZ,DT,DL2,MAXIT,EPS,CMEGA,
85 CARBP,TAUX,TAUY,TTCT,MAR,MRH,IN,JN,IN,JN)
86 CALL PRUV(I,J,K,IN,JN,KN,U,V)
87 6 CONTINUE
88 CALL PREROR(IWN,IW,WHLDT,JW,JWN,IN,JN)
89 CALL PRITE(I,J,IW,JW,XINT,YINT,IN,JN)
90 CALL PRSOPC(IW,JW,IWN,JWN,FH)
91 CALL POPSXY(I,J,IN,JN,DPSX,CPSY)
92 CALL PRITEX(ITN,EX)
93 CALL PRPINT(IW,JW,IWN,JWN,P)
94 CALL PRUV(I,J,K,IN,JN,KN,U,V)
95 CALL PRWH(IW,JW,K,IWN,JWN,KN,WRH)
96 END

```

### 9.1.3 TMAIN1 (Main Program for Near Field)

This is a main program and is used for obtaining velocity and temperature distribution in the near field of a constant depth basin. The parameter statement defines the size of the computational domain. The subroutine READ 3 reads MAR and MRH matrices. The subroutine INITIA initializes the velocity and pressure field. The subroutine INITIT initializes the temperature field. It sets temperature in the whole domain equal to the reference temperature. The Flow Chart which the main program follows is shown in Fig. ( 9.1). The subroutines are also given in the Flow Chart.

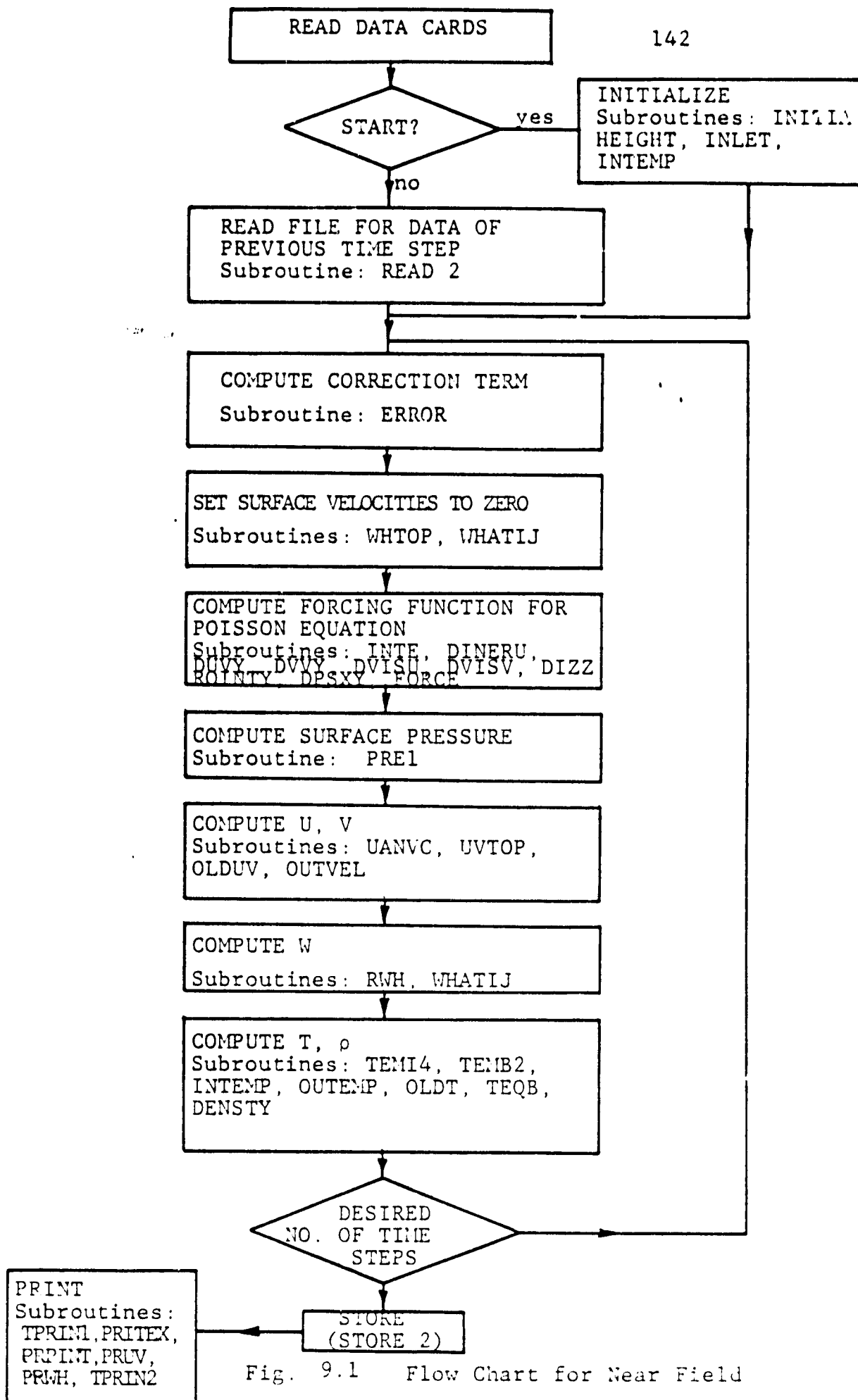


Fig. 9.1 Flow Chart for Near Field

**DATE** \_\_\_\_\_ **INITIALS** \_\_\_\_\_

```

57      DO 6 LL=1,LLN
58      DO 5 L=1,LN
59      TTOT=TTOT+DT
60      CALL ERROR(IWN,JWN,IN,JW,DT,WH,WHLOT,KN,MRH)
61      CALL WHTCP(IN,JW,IWN,JWN,KN,WH,K,MRH)
62      CALL WHATIJ(I,J,K,IN,JN,IN,JN,KN,IWN,JWN,W,WH,MAR)
63      CALL INTE(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3,AI,
64      CAH,AV,TAUX,TAUY,DX,DY,DZ,C,E,DT,DPSX,DPSY,AP)
65      CALL ROINTX(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,
66      CHAR,RINTX,HX,XINT)
67      CALL ROINTY(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,MAR,
68      CRINTY,HY,YINT)
69      CALL DPSXY(I,J,IN,JN,IN,JW,IWN,JWN,DPSX,DPSY,P,DX,DY,MAR)
70      CALL FORCE(I,J,IN,JN,XINT,YINT,WHLOT,DX,DY,HI,HX,HY,MRH,
71      CDPSX,DPSY,FH,AP,IN,JN,IWN,JWN)
72      CALL PREZIEPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FH,DL2,OMEGA,
73      CMRH,I,J,K,IN,JW,DX,DY,EX,IWN,JWN,ARBP)
74      CALL UVT(I,J,K,IN,JW,IN,JN,KN,IWN,JWN,U,V,D,E,H,G,DX,DY,DZ,
75      CRINTX,RINTY,EUL,W,DT,AI,AP,AH,AV,A3,HI,HX,HY,P,MAR)
76      CALL UVTOP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
77      CALL OUTVEL(I,J,K,IN,JN,KN,H,G)
78      CALL OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
79      CALL OLDUV(I,J,K,IN,JN,KN,H,G,U,V)
80      CALL RWH(I,J,K,IN,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,DX,DY,DZ,MRH)
81      CALL WHATIJ(I,J,K,IN,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
82      DO 20 I=1,IN
83      DO 20 J=1,JN
84      WD(I,J,1)=W(I,J,1)
85      20 CONTINUE
86      DO 30 I=1,IN
87      DO 30 J=1,JN
88      W(I,J,1)=0.0
89      30 CONTINUE
90      CALL TEMI4(I,J,K,IN,JN,KN,U,V,T,TD,CX,
91      CCB,
92      CDY,DZ,W,DT,TAI,TAH,TAV,B3,HI,HX,HY,MAR,AKT,TREF,TAMB)
93      CALL TEMB2(I,J,K,IN,JN,KN,TD,DX,DY,DZ,MAR,CB,HI,AKT,CW,TAMB,
94      CHX,HY,T,TPEF,TAV,TAI,TAH,B3,DT)
95      CALL INTEPP(I,J,K,IN,JN,KN,T,TD,TLL,TMM)
96      CALL OUTEMP(I,J,K,IN,JN,KN,TD)
97      CALL OLD T(I,J,K,IN,JN,KN,T,TP)
98      CALL OLD T(I,J,K,IN,JN,KN,TD,T)
99      CALL TEQB(I,J,K,IN,JN,KN,T,MAR)
100     CALL DENSITY(I,J,K,IN,JW,IN,JN,KN,IWN,JWN,A,B,C,MAR,MRH,T,TW,
101     CRC,RCW,RREF,TREF)
102     DO 40 I=1,IN
103     DO 40 J=1,JN
104     W(I,J,1)=WD(I,J,1)
105     40 CONTINUE
106     5 CONTINUE
107     CALL RWRH(I,J,K,IN,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,HX,HY,
108     CDX,DY,DZ,MRH,WRH)
109     CALL RWR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,HY,DZ,MAR)
110     CALL STORE2(U,V,WH,P,I,J,K,IN,JN,KN,IWN,JWN,D,E,HX,HY,
111     CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WPH,TAI,TAH,
112     CTAV,AKT,CB,CW,A,B,C,EUL,T,TW,RC,RCW,TE,PREF,TREF,TO,TAMB,TTOT)
113     CALL PRPAPA(AT,AH,AV,AP,DX,DY,DZ,DT,DL2,MAXIT,EPS,OMEGA,

```

```
114 CARBP,TAUX,TAUY,TTOT,MAR,MRH,IN,JN,IWN,JWN)
115 CALL TPRIN1(TAI,TAM,TAV,CB,CW,AKT,TREF,RREF,EUL,A,B,C,TE,TO)
116 CALL PRITEX(I,TN,EX)
117 CALL PRPINT(IW,JW,IWN,JWN,P)
118 CALL PRUV(I,J,K,IN,JN,KN,U,V)
119 CALL PRWH(IW,JW,K,IWN,JWN,KN,WRH)
120 CALL TPRIN2(I,J,K,IN,JN,KN,T,RO,TREF)
121 6 CONTINUE
122 END
```



#### 9.1.4 TMAIN2 (Main Program for Near Field)

This is the same as TMAIN 1. except, this program is to be used when the depth is variable. The subroutine GRADSL computes slopes of the bottom in x and y directions respectively.

N\*DOC. TMAIN2

```

1      PARAMETER IN=13,JN=20,KN=5,IN=17,JWN=19
2      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IN,JN,KN),
3      CHR(IN,JN,KN),WRH(IN,JN,KN),P(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
4      CMHDT(IN,JN),XINT(IN,JN),YINT(IN,JN),H(I,JN,KN),G(IN,JN,KN),
5      CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IN,JN),FH(IN,JN),
6      COPSX(IN,JN),OPSY(IN,JN)
7      DIMENSION A3(KN)
8      DIMENSION T(IN,JN,KN),TP(IN,JN,KN),TD(IN,JN,KN),RO(IN,JN,KN),
9      CRINTX(IN,JN,KN),RINTY(IN,JN,KN),WD(IN,JN,KN)
10     DIMENSION TW(IN,JN,KN),ROW(IN,JN,KN),X(JN),II(IN)
11     INM1=IN-1
12     READ 1, IRUN
13     READ 1, LN
14     READ 1, LLN
15     1   FORMAT (I5)
16     READ 2, VVIS,ABF
17     A3(1)=VVIS
18     A3(2)=VVIS
19     A3(3)=VVIS
20     A3(4)=VVIS
21     A3(5)=VVIS
22     B3=VVIS
23     READ 2, AI,AH,AV,AP
24     READ 2, EPS,MAXIT,OMEGA,ARBP
25     READ 2, DX,DY,DZ
26     READ 2, TAI,TAH,TAV
27     READ 2, A,B,C
28     READ 2, TC
29     READ 2, AKT,EUL,CW,CB
30     READ 2,AA,BB,CC
31     READ 2,TLL,TMM
32     2   FORMAT (I)
33     DL2=DX*DX
34     TREF=TO
35     RREF=A*B*TO*C*TO*TO
36     IF(IRUN.GT.0) GO TO 3
37     CALL READ3(I,J,IN,JN,IW,JW,INW,JWN,MAR,MRH)
38     CALL INITIA(IN,JN,KN,INW,JWN,U,V,W,WH,D,E,
39     CP,I,J,K,IN,JW,AFBP)
40     CALL INITIT(I,J,K,IN,JN,KN,INW,JW,INW,JWN,A,B,C,T,RO,MAR,MRH,TREF,
41     CRREF,TW,ROW,TC)
42     CALL IRDATA(I,J,K,IN,JN,KN,T,TREF,II)
43     CALL HEIGH1(I,J,K,IN,JN,KN,HI,HX,HY,CC,JX)
44     CALL GRADS1(IN,JN,KN,INW,JWN,HI,HX,HY,MAR,MRH,DX,DY)
45     CALL INLET(I,J,K,IN,JN,KN,V,G,AA,BB)
46     CALL INTMP(I,J,K,IN,JN,KN,T,TO,TLL,TMM)
47     CALL CURNT(I,J,K,IN,JN,KN,U,V,D,E,H,G)
48     GO TO 4
49     3   CONTINUE
50     CALL READ2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,INW,JWN,D,E,HX,HY,HI,
51     CMAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT
52     C,CB,CW,A,B,C,EUL,T,TH,RO,ROW,TE,RREF,TREF,TO,TAMB,ITOT)
53     CALL INLET(I,J,K,IN,JN,KN,V,G,AA,BB)
54     CALL INTMP(I,J,K,IN,JN,KN,T,TO,TLL,TMM)
55     4   CONTINUE
56     READ 2, TAMB

```

```

57      TE=(TAMB-TREF)/TREF
58      READ 2,TAUX,TAUY
59      READ 2, DT
60      DO 6 LL=1,LLN
61      DO 5 L=1,LN
62      TTOT=TTOT+DT
63      CALL ERRCR(IWN,JWN,IW,JW,DT,WH,WHLOT,KN,MRH)
64      CALL WHTOP(IW,JW,IW,JW,KN,WH,K,MRH)
65      CALL WHATI(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
66      CALL INTE(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3,AI,
67      CAH,AV,TAUX,TAUY,DX,DY,DZ,D,E,DT,DPSX,DPSY,AP)
68      CALL CORINT(I,J,K,IN,JN,KN,ABR,U,V,XINT,YINT,DZ,HI,MAR)
69      CALL ROINTX(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,
70      CMAR,RINTX,HX,XINT)
71      CALL ROINTY(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,MAR,
72      CRINTY,HY,YINT)
73      CALL DPSXY(I,J,IN,JN,IW,JW,IWN,JWN,DPSX,DPSY,P,DX,DY,MAR)
74      CALL FORCE(I,J,IW,JW,XINT,YINT,WHLOT,DX,DY,HI,HX,HY,MRH,
75      CDPSX,DPSY,FH,AP,IN,JN,IWN,JWN,RINTX,RINTY,U,V,EUL,APR,MAR,KN)
76      CALL PREC(EPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FH,CL2,OMEGA,
77      CMRH,I,J,K,IW,JW,DX,DY,EX,IWN,JWN,ARPP)
78      CALL UVT(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,D,E,H,G,DX,DY,DZ,
79      CRINTX,RINTY,EUL,W,DT,AI,AP,AH,AV,A3,HI,HX,HY,P,MAR)
80      CALL UVTCP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
81      CALL OTVELS(I,J,K,IN,JN,KN,H,G)
82      CALL OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
83      CALL OLDUV(I,J,K,IN,JN,KN,H,G,U,V)
84      CALL RWH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,DX,DY,DZ,MRH)
85      CALL WHATI(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
86      15 CONTINUE
87      DO 20 I=1,IN
88      DO 20 J=1,JN
89      WD(I,J,1)=W(I,J,1)
90      20 CONTINUE
91      DO 30 I=1,IN
92      DO 30 J=1,JN
93      W(I,J,1)=0.0
94      30 CONTINUE
95      CALL TEMI4(I,J,K,IN,JN,KN,U,V,T,TC,EX,
96      CCB,
97      CDY,DZ,W,DT,TAI,TAH,TAV,B3,HI,HX,HY,MAR,AKT,TREF,TAMP)
98      CALL TEMB2(I,J,K,IN,JN,KN,TD,DX,DY,CZ,MAR,CB,HI,AKT,CW,TAMB,
99      CHX,HY,T,TREF,TAV,TAI,TAH,B3,DT)
100     CALL INTEMP(I,J,K,IN,JN,KN,T,TD,TLL,TMM)
101     CALL OUTEMP(I,J,K,IN,JN,KN,TD)
102     CALL OLD T(I,J,K,IN,JN,KN,T,TP)
103     CALL OLD T(I,J,K,IN,JN,KN,TD,T)
104     CALL TEQB(I,J,K,IN,JN,KN,T,MAR)
105     CALL DENSITY(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,A,B,C,MAR,MRH,T,TW,
106     CRO,ROW,PREF,TREF)
107     DO 40 I=1,IN
108     DO 40 J=1,JN
109     W(I,J,1)=WD(I,J,1)
110     40 CONTINUE
111     5 CONTINUE
112     CALL RWRH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,HX,HY,
113     CDX,DY,DZ,MRH,WRH)

```

```

114 CALL RWR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,MY,DZ,MAR)
115 CALL STORE2(U,V,W,H,P,I,J,K,IN,JN,KN,IWN,JWN,C,E,HX,MY,
116 CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,
117 CTAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAMB,TTOT)
118 CALL PRPARA(AI,AH,AV,AP,DX,DY,DZ,DT,CL2,MAXIT,EPS,OMEGA,
119 CARBP,TAUX,TAUY,TTOT,MAR,MRH,IN,JN,IWN,JWN)
120 CALL INTEMP(I,J,K,IN,JN,KN,T,TD,TLL,TMH)
121 CALL CALMS(I,J,K,IN,JN,KN,U,V,D,E,H,G)
122 CALL TPRIN1(TAI,TAH,TAV,CB,CW,AKT,TREF,RREF,EUL,A,B,C,TE,TO)
123 CALL PRITEX(I,TN,EX)
124 CALL PRPINT(IW,JW,IWN,JWN,P)
125 CALL PRUV(I,J,K,IN,JN,KN,U,V)
126 CALL PRWH(IW,JW,K,IWN,JWN,KN,WRH)
127 CALL TPRIN2(I,J,K,IN,JN,KN,T,RO,TREF)
128 CONTINUE
129 END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

CONFIDENTIAL

MAIN PROGRAMS FOR FAR-FIELD

 *Handwritten signature*

#### 9.1.5 TMAIN4 (Main Program for Far-Field)

This is a main program. This program is used for initializing velocity and temperature fields for shallow unstratified basins with constant vertical viscosity. The program also fills in the depth matrix. The program fills in the MAR and MRH matrices which define the relative location of computational points in the full and the half grid systems, respectively. The PARAMETER statement defines the size of the computational grid system. The program uses the data element INDATA. First, twelve lines of the data are read in by TMAIN4. The subroutine READ3 reads the MAR and MRH matrices in that sequence. The INITIA subroutine initializes the velocity and pressure fields. The velocity field is set equal to zero and the pressure field is set equal to unity everywhere. The subroutine INITIT initializes the temperature field equal to the reference temperature everywhere in the domain of interest. The subroutine HITEA reads in the depth matrix. The subroutine GRADS computes slopes of the bottom of the basin in x and y directions. The print statements are included in subroutines READ3A and GRADS. The subroutine READ3A prints out MAR and MRH matrices. The subroutine GRADS prints out the depth matrix and the two matrices of x and y slopes. The program stores initialized and computed physical quantities on Unit 8. Element RTM4 is used to provide computer commands necessary to execute this program on UNIVAC-1106 computer.

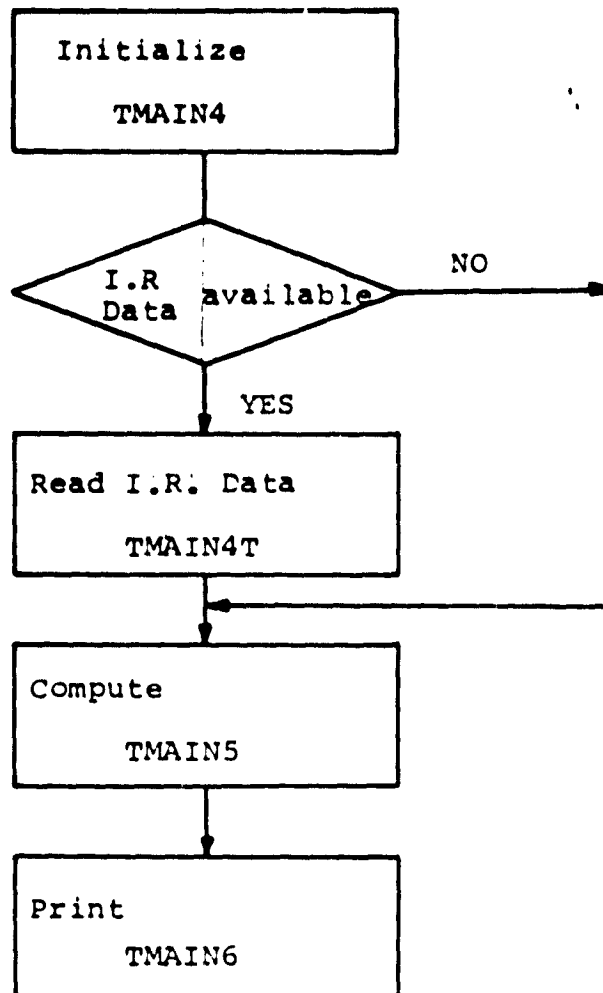
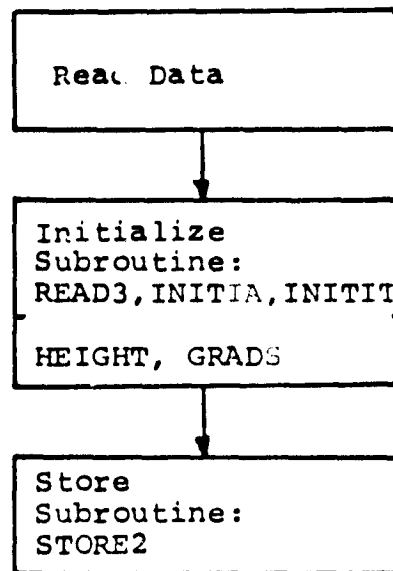


Fig. 9.2 Flow Chart

## TMAIN4



## TMAIN5

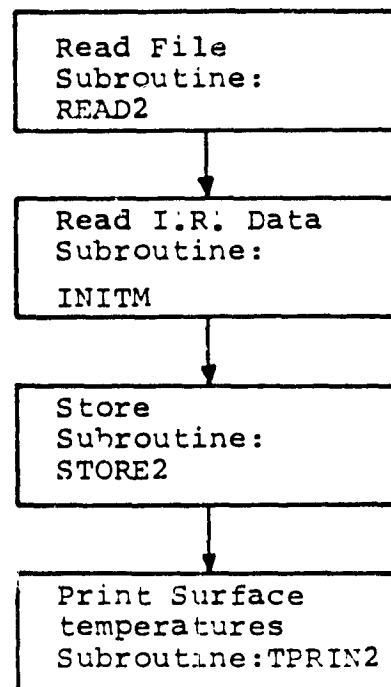


Fig. 9.3 Flow Chart



## TMAIN5

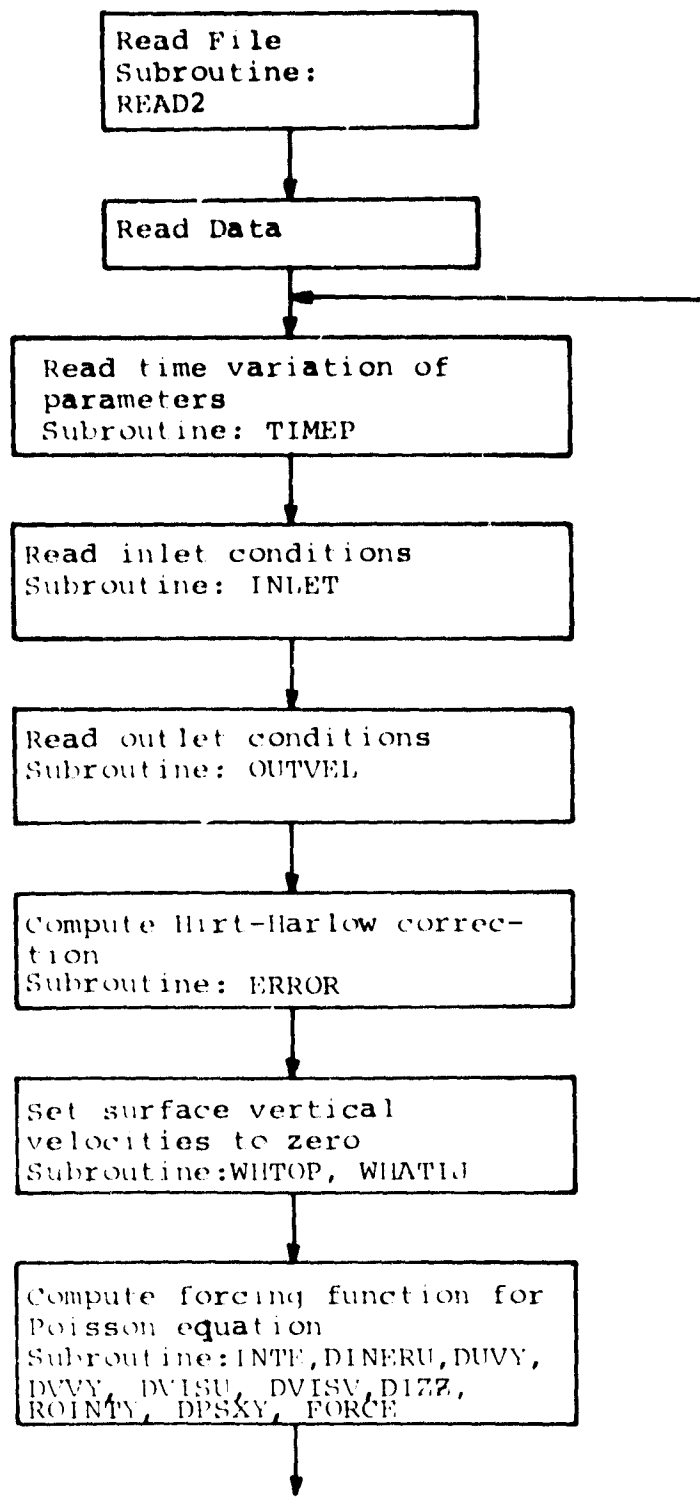


Fig. 9.4 Flow Chart

(Continued on next page)

## TMAIN5 continued

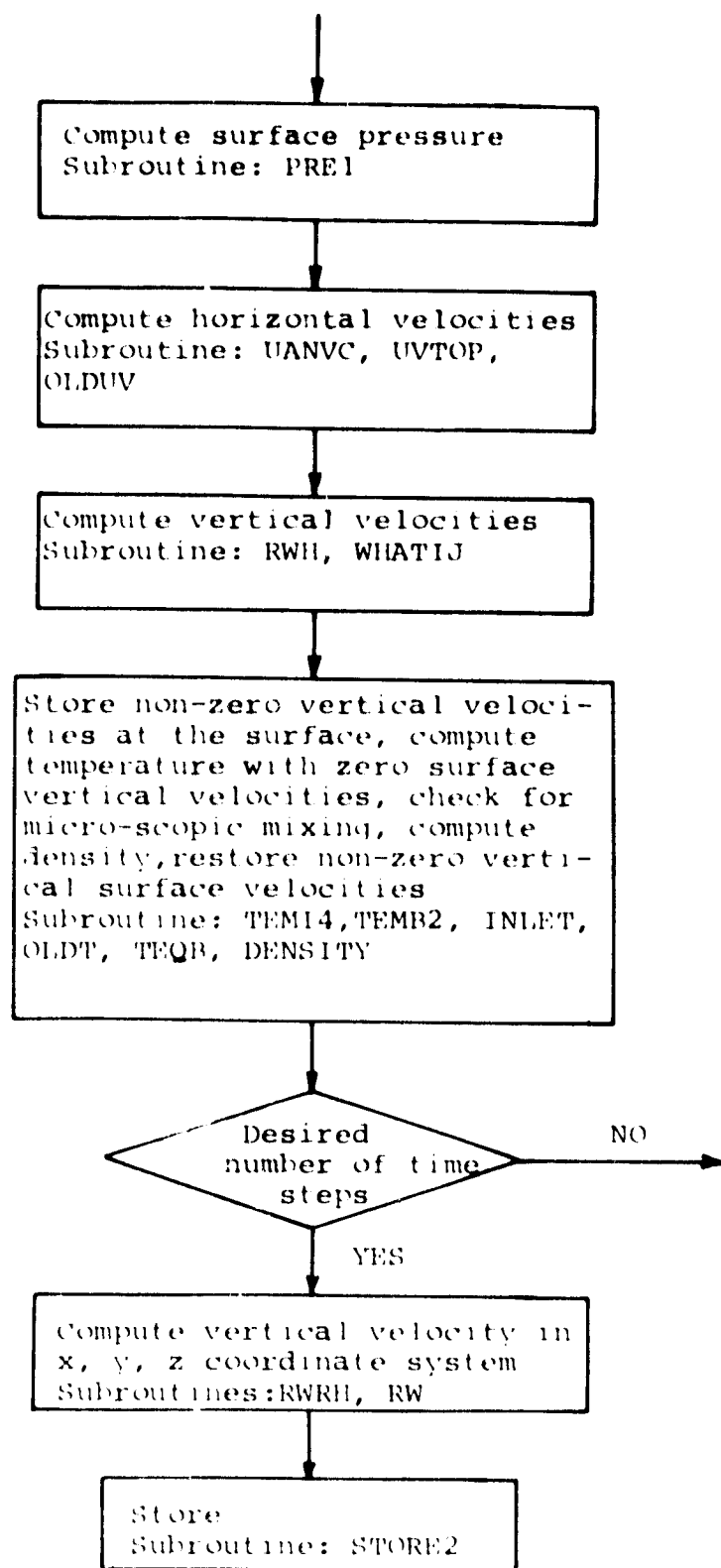


Fig. 15 Flow Chart (contd)

H\*TPFS.TPAIN4

```

1      PARAMETER IN=29,JN=13,KN=6,IWN=28,JWN=12
2      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
3      CWR(IN,JN,KN),LRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
4      CWHLDT(IWN,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
5      CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN),FH(IWN,JWN),
6      CCPSX(IN,JN),DPSY(IN,JN)
7      DIMENSION A3(KN)
8      DIMENSION T(IN,JN,KN),TD(IN,JN,KN),RO(IN,JN,KN),
9      CRINTX(IN,JN,KN),RINTY(IN,JN,KN),WD(IN,JN,KN)
10     INM=IN-1
11     READ 1, IRUN
12     READ 1, LN,LLN
13     1    FORMAT (16I5)
14     READ 1,NOTGR
15     READ 2, VVIS,ABR
16     A3(1)=VVIS
17     A3(2)=VVIS
18     A3(3)=VVIS
19     A3(4)=VVIS
20     A3(5)=VVIS
21     A3(6)=VVIS
22     B3=VVIS
23     READ 2, AI,AH,AV,AP
24     READ 2, EPS,MAXIT,OMEGA,AREP
25     READ 2, DX,DY,DZ
26     READ 2,CC
27     READ 2,TAUXM,TALYM
28     READ 2,DT
29     READ 2, TAI,TAH,TAV
30     READ 2, A,B,C
31     READ 2, TO
32     READ 2, AKT,EUL,CW,CB
33     READ 2,TAMB,TINM
34     READ 2,IIN,JIA,IOUT,JOUT,UINM,VINM
35     2    FORMAT (I)
36     DL2=DX*DX
37     TREF=TO
38     RREF=A+B*TO+C*TC*TO
39     IF (IRUN.EQ.0) GO TO 15
40     IF (IRUN.EQ.1) GO TO 16
41     IF (IRUN.EQ.2) GO TO 3
42     15    CONTINUE
43     GO TO 10CC
44     REWIND 9
45     READ (9) ((MAR(I,J),I=1,IN),J=1,JN),
46     C((MRH(IW,JW),IW=1,IWN),JW=1,JWN),
47     C((HI(I,J),I=1,IN),J=1,JN),
48     C((HX(I,J),I=1,IN),J=1,JN),
49     C((HY(I,J),I=1,IN),J=1,JN)
50     REWIND 9
51     10CC  CONTINUE
52     CALL READ3(I,J,IN,JN,IW,JW,IWN,JWN,MAR,MRH)
53     CALL INITI(I,IN,JN,KN,IWN,JWN,U,V,W,WH,C,E,P,I,J,K,IW,JW,AREP)
54     CALL INITIT(I,J,K,IN,JN,KN,IW,JW,IWN,JWN,A,B,C,I,RO,MAR,MRH,TREF,
55     CRREF,TO)
56     CALL HEIGHT(I,J,K,IN,JN,KN,HI,HX,HY,CC)

```

```

57      CALL GRAD5(IN,JA,KN,IWN,JWN,HI,HX,HY,MAR,MRH,DX,DY)
58      TTOT=C.O
59      ITN=O
60      EX=C.
61      GC TO 4
62      16  CONTINUE
63      3   CONTINUE
64      4   CONTINUE
65      CALL STORE2(U,V,W,H,P,I,J,K,IM,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,
66      CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,
67      CTAV,AKT,CB,CW,A,B,C,EUL,T,RO,TE,RPEF,TREF,TO,TAMB,TTOT,
68      CITH,EX)
69      CALL TPRIN9(I,J,K,IN,JN,KN,T,RO,TREF,MAR)
70      6   CONTINUE
71      END

```

#### 9.1.6 TMAIN4T (Main Program for Far Field)

This is a main program. This program is used for shallow unstratified basins with constant vertical viscosity. This program is used to update the initial temperature field from a constant value everywhere as defined by TMAIN4 to a better defined temperature field. The subroutine READ2A reads in the values of physical quantities stored by TMAIN4. The read-in unit is numbered 7. The subroutine INITM reads in the surface temperatures defined as data element ITPK1. Such temperatures are obtained from infrared scanning of the water surface. The temperatures below the surface are computed from a specified temperature gradient in the vertical direction. Computations for temperatures below the surface are also made in subroutine INITM. The subroutine STOR2A writes out the updated values on unit #8. The subroutine TPRIN2 prints out the updated temperature field. The program also sets the time count to zero. The element RTM4T provides with the computer commands necessary to execute program TMAIN4T on UNIVAC-1106 computer.

10TPFS.TMAIN4T

```

1  PARAMETER IN=29,JN=13,KN=6,IW=28,JWN=12
2  PARAMETER NTL=14,NTLV=2
3  DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IW,JWN,KN),
4  CWR(IN,JN,KN),MRH(IW,JWN,KN),P(IW,JWN),D(IN,JN,KN),E(IN,JN,KN),
5  CWHLOT(IW,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
6  CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IW,JWN),FH(IW,JWN),
7  CDPSX(IN,JN),DFSY(IN,JN)
8  DIMENSION A3(KN),UV(JN),THETA(JN),AMINT(NTL,NTLV)
9  DIMENSION T(IN,JN,KN),TO(IN,JN,KN),PO(IN,JN,KN),
10 CRINTX(IN,JN,KN),RINTY(IN,JN,KN),RO(IN,JN,KN)
11 3 CONTINUE
12 CALL READ2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IW,JWN,D,E,HX,HY,HI,
13 CHAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,MRH,TAI,TAH,TAV,AKT
14 C,CB,CW,A,B,C,EUL,T,RO,TE,PRF,TREF,TO,TAMB,TTOT,ITN,EX)
15 4 CONTINUE
16 READ 2,((AMINT(N,L),L=1,NTLV),N=1,NTL)
17 2 FORMAT (
18 CALL INITIT(I,J,K,IN,JN,KN,IW,JW,IW,JWN,A,B,C,T,RO,MAR,MRH,TREF,
19 CRREF,TO,AMINT,HI,NTL,NTLV)
20 CALL INITM(I,J,K,IN,JN,KN,IW,JW,IW,JWN,A,B,C,T,RO,MAR,MRH,TREF,
21 CRREF,TO,HI)
22 TTOT=0.000
23 CALL STORE2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IW,JWN,D,E,HX,HY,
24 CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,T,UY,W,WR,MRH,TAI,TAH,
25 CTAV,AKT,CB,CW,A,B,C,EUL,T,RO,TE,RRF,TREF,TO,TAMB,TTOT,
26 CIITN,EX)
27 CALL TPRIN9(I,J,K,IN,JN,KN,T,RO,TREF,MAR)
28 6 CONTINUE
29 END

```

### 9.1.7 TMAIN5 (Main Program for Far Field)

This is a main program. This is the program which performs most of the computations. Solutions are propagated in time for both the velocity field and the temperature field. Values stored by TMAIN4T are read in by subroutine READ2A from Unit #7. First twelve lines of data element INDATA (defined as data element INDATA5) are used to provide with the values for basic parameters. The subroutine INLETA imposes the inlet and outlet velocities and temperatures onto the computational domain. INLETA subroutine reads lines 13 through the last line of data element INDATA5. The subroutine ERROR computes the contribution of nonzero surface vertical velocities to the rigid lid pressure. The subroutines WHTOP and WHATIJ set the surface vertical velocities to zero. The subroutines INTE partially evaluates the forcing function in the Poisson's equation for rigid-lid pressure. The subroutine CORINT adds the contribution of Coriolis force to the values computed by INTE. The subroutines ROINTX and ROINTY add the contribution of bouyancy to the terms in the forcing function. The subroutine DPSXY computes the pressure gradients at the surface along the boundary for use in the solution of Poisson's equation for rigid lid pressure. The subroutine FORCE combines all terms of the forcing function. The subroutine PRE1 computes the rigid lid pressure field by iteration. The subroutine UVT uses the  $u$  and  $v$  momentum equations to evaluate  $u$  and  $v$  at grid points below the surface. The subroutine UANVC add the contribution of Coriolis force to  $u$  and  $v$ . The subroutine UVTOP computes the surface velocities from the velocities below the surface by

utilizing the specified velocity gradient due to wind. The subroutine OLDUV updates the values of  $u$  and  $v$ . The subroutines RWH and WHATIJ compute the vertical velocity, the continuity equation is used in the process. The nonzero surface vertical velocities are then saved as WD. The surface vertical velocities  $W$  are set equal to zero before going into the energy equation. The subroutine TEMI4 computes temperatures at the interior points. The subroutine TEMB2 computes temperatures at the boundary points. The subroutine OLDT updates the values of temperature. The subroutine TEQB checks for thermal instability and mixes the water to create stable temperature field. The subroutine DENSTY computes the new density field from the new temperature field using the equation of state. The nonzero vertical surface velocities are now restored. The subroutines RWRH and RWR compute the vertical velocities in the  $x$ - $y$ - $z$  coordinates from the values in  $\alpha$ ,  $\beta$ ,  $\gamma$  coordinates. The subroutine STOR2A stores the updated values on unit #8. The element RTM5 provides, the computer commands, necessary to execute program TMAIN5 on UNIVAC-1106 computer.



## HOTPFS.TMAINE

```

1  PARAMETER IN=29,JN=13,KN=6,ILN=28,JWN=12
2  DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
3  CLR(IN,JN,KN),MRH(ILN,JLN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
4  CWHLD(IWA,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
5  CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN),FH(ILN,JLN),
6  CDPSX(IN,JN),DPSY(IA,JN)
7  DIMENSION A3(KN)
8  DIMENSION T(IN,JN,KN),TD(IN,JN,KN),RO(IN,JN,KN),
9  CRINTX(IN,JN,KN),RINTY(IN,JN,KN),WC(IN,JN,KN)
10 CALL READ2(U,V,W,P,I,J,K,IL,JW,IN,JN,KN,IWN,JWN,C,E,HX,HY,HI,
11 CHAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT
12 C,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,ITN,EX)
13 INM=IN-1
14 READ 1, IPUN
15 READ 1, LN,LLN
16 1 FORMAT (1615)
17 READ 1,NDTGR
18 READ 2, VVIS,ABF
19 A3(1)=VVIS
20 A3(2)=VVIS
21 A3(3)=VVIS
22 A3(4)=VVIS
23 A3(5)=VVIS
24 A3(6)=VVIS
25 B3=VVIS
26 READ 2, AI,AH,AV,AP
27 READ 2, EPS,MAXIT,OMEGA,AFBP
28 READ 2, DX,DY,DZ
29 READ 2,CC
30 READ 2,TAUXM,TAUYM
31 READ 2,DT
32 READ 2, TAI,TAH,TAV
33 READ 2, A,B,C
34 READ 2, TO
35 READ 2, AKT,EUL,CW,CB
36 READ 2,TAMB,TINM
37 READ 2,IIN,JIN,IOUT,JOUT,UINM,VINM
38 2 FORMAT (1)
39 DL2=DX*DX
40 TREF=TO
41 RREF=A*B*TO+C*TC*TO
42 4 CONTINUE
43 TE=(TAMB-TREF)/TREF
44 DO 5 L=1,LN
45 TTOT=TTOT*DT
46 CALL TIMEP(TAMB,AKT,UINM,VINM,TINM,TAUXM,TAUYM,TTOT,DT)
47 CALL INLET(I,J,K,IN,JN,KN,U,V,W,H,G,TTOT,DT,
48 CNDTGR,TAUXM,TAUYM,TAUX,TAUY,TINM,T,IIN,JIN,UINM,VINM)
49 CALL OUTVEL(I,J,K,IN,JN,KN,U,V,W,H,G,MAR,IOUT,JOUT,IIN,JIN)
50 CALL ERROR(IWN,JWN,IL,JW,DT,WH,WHLDT,KN,MRH)
51 CALL WHTCP(IL,JW,IWN,JWN,KN,WH,K,MRH)
52 CALL WHATIJ(I,J,K,IL,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
53 CALL INTE(I,J,K,IL,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3,AI,
54 CAH,AV,TAUX,TAUY,DX,DY,DZ,C,E,DT,CDPSX,DPSY,AP,T,TREF)
55 CALL CORINT(I,J,K,IN,JN,KN,AFB,U,V,XINT,YINT,DZ,HI,MAR)
56 CALL ROINTX(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,

```

```

57 CHAR,RINTX,HX,XINT)
58 CALL ROINTY(I,J,K,IN,JN,KN,CX,DY,DZ,RO,AP,EUL,HI,MAR,
59 CRINTY,HY,YINT)
60 CALL DPSXY(I,J,IN,JN,IW,JW,IWN,JWN,CPSX,CPSY,P,DX,DY,MAR)
61 CALL FORCE(I,J,IW,JW,XINT,YINT,WHLOT,CX,DY,HI,HX,HY,MRH,
62 CDPSX,DPSY,FH,AP,IN,JN,IWN,JWN,RINTX,RINTY,U,V,EUL,ABR,MAR,KN)
63 CALL PRE1(EPS,MAXIT,IN,JN,P,ITN,DPSX,CPSY,FH,CL2,OMEGA,
64 CMRH,I,J,K,IW,JW,DX,DY,EX,IWN,JWN,ARBP)
65 CALL UVT(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,D,E,H,G,DX,DY,DZ,
66 CRINTX,RINTY,ELL,B,DT,AI,AF,AH,AV,AJ,HI,HX,HY,P,MAR,T,TREF)
67 CALL UANVC(I,J,K,IN,JN,KN,AFR,DT,U,V,H,G,HI,MAR)
68 CALL UVTCP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
69 CALL INLET(I,J,K,IN,JN,KN,U,V,H,G,TTOT,DT,
70 CNDTGR,TAUXM,TAUYM,TAUX,TAUY,TINM,T,IIN,JIN,UINM,VINM)
71 CALL OUTVEL(I,J,K,IN,JN,KN,U,V,H,G,MAR,ICUT,JCUT,IIN,JIN)
72 CALL OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
73 CALL OLDUV(I,J,K,IN,JN,KN,H,G,U,V)
74 CALL RWRH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,CX,DY,DZ,MRH)
75 CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
76 DO 20 I=1,IN
77 DO 20 J=1,JN
78 WD(I,J,1)=W(I,J,1)
79 20 CONTINUE
80 DO 30 I=1,IN
81 DO 30 J=1,JN
82 W(I,J,1)=0.0
83 30 CONTINUE
84 CALL TEMI4(I,J,K,IN,JN,KN,U,V,T,TC,CX,CB,DY,DZ,W,DT,TAI,TAH,TAV,
85 CD3,HI,HX,HY,MAR,AKT,TREF,TAMB)
86 CALL TEMB2(I,J,K,IN,JN,KN,TC,DX,DY,CZ,MAR,CB,HI,AKT,CW,TAMB,HX,
87 CHY,T,TREF,TAV,TAI,TAH,B3,DT)
88 CALL OLD1(I,J,K,IN,JN,KN,TD,T)
89 CALL TSTAR(I,J,K,IN,JN,KN,T,MAR)
90 CALL DENSITY(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,A,B,C,MAR,MRH,T,
91 CRO,RREF,TREF)
92 DO 40 I=1,IN
93 DO 40 J=1,JN
94 W(I,J,1)=WD(I,J,1)
95 40 CONTINUE
96 5 CONTINUE
97 CALL RWRH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,HX,HY,
98 CDX,DY,DZ,MRH,WRH)
99 CALL RWR(I,J,K,IN,JN,KN,U,V,W,WP,HI,HX,HY,DZ,MAR)
100 CALL STORE2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,
101 CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,
102 CTAV,AKT,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,
103 CITH,EX)
104 END
IT,5

```

#### 9.1.8 TMAIN5T (Main Program for Far Field)

This program simulates temperatures only for unstratified shallow basins with constant vertical eddy viscosity. Steps involved in its execution are similar to those involved in the execution of TMAIN5. This program uses data element INDATA5.

SHOTPFS.YMAINE1

```

1  PARAMETER IN=29,JN=13,KN=6,IN=28,JN=12
2  DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IN,JN,KN),
3  CUR(IN,JN,KN),WRH(IN,JN,KN),P(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
4  CH(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IN,JN),
5  DIMENSION G(IN,JN,KN),H(IN,JN,KN)
6  DIMENSION A3(KN)
7  DIMENSION T(IN,JN,KN),TO(IN,JN,KN),RO(IN,JN,KN),WD(IN,JN,KN)
8  CALL READ2(U,V,W,H,P,I,J,K,IN,JN,IN,JN,KN,IN,JN,KN,D,E,HX,HY,HI,
9  CHAK,MRH,AT,AH,AV,AP,CX,DY,CZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT
10 C,CB,CW,A,B,C,EUL,T,PC,IE,RREF,TREF,TO,TAMB,TTOT,ITN,EX)
11 INM1=IN-1
12 READ 1, IRUN
13 READ 1, LN,LLN
14 1 FORMAT (16I5)
15 READ 1,NDTGR
16 READ 2, VVIS,ABR
17 A3(1)=VVIS
18 A3(2)=VVIS
19 A3(3)=VVIS
20 A3(4)=VVIS
21 A3(5)=VVIS
22 A3(6)=VVIS
23 B3=VVIS
24 READ 2, AI,AH,AV,AP
25 READ 2, EPS,MAXIT,OMEGA,ARBF
26 READ 2, DX,DY,DZ
27 READ 2,CC
28 READ 2,TAUXM,TAUYM
29 READ 2,DT
30 READ 2, TAI,TAH,TAV
31 READ 2, A,B,C
32 READ 2, TO
33 READ 2, AKT,EUL,CW,CB
34 READ 2,TAMB,TINP
35 READ 2,IIN,JIN,IOUT,JOUT,UINM,VINM
36 2 FORMAT (I)
37 DL2=DX*DX
38 TREF=TO
39 RREF=A*B*TO+C*TO*TO
40 4 CONTINUE
41 DO 20 I=1,IN
42 DO 20 J=1,JN
43 W(I,J,1)=W(I,J,1)
44 2C CONTINUE
45 DO 30 I=1,IN
46 DO 30 J=1,JN
47 W(I,J,1)=C.0
48 3C CONTINUE
49 DO 6 LL=1,LLN
50 TTOT=TTOT+DT
51 CALL TIMEP(TAMB,AKT,UINM,VINM,TINP,TAUXM,TAUYM,TTOT,DT)
52 CALL INLET(I,J,K,IN,JN,KN,U,V,W,H,G,TTOT,DT,
53 CNDIGR,TAUXM,TAUYM,TAUX,TAUY,TINP,T,IIN,JIN,UINM,VINM)
54 CALL OUTVEL(I,J,K,IN,JN,KN,U,V,W,H,G,MAR,IOUT,JOUT,IIN,JIN)
55 CALL TEMIN(I,J,K,IN,JN,KN,U,V,T,TD,CX,CB,DY,DZ,W,DT,TAI,TAH,TAV,
56 CB3,HI,HX,HY,MAR,AKT,TREF,TAMB)

```

```

57      CALL TLMB2(I,J,K,IN,JN,KN,TO,DX,DY,CZ,MAR,CB,MI,AKT,CW,TAMB,HX,
58      CHY,I,TREF,YAV,TAI,TAM,B3,CT)
59      CALL OLD T(I,J,K,IN,JN,KN,TO,T)
60      CALL TSTAE(I,J,K,IN,JN,KN,T,MAR)
61      6      CCNTINUE
62      CALL DENSITY(I,J,K,IW,JW,IN,JN,KN,IMN,JMN,A,B,C,HAP,PRH,T,
63      CRO,RREF,TREF)
64      DO 40 I=1,IN
65      DO 40 J=1,JN
66      W(I,J,1)=BU(I,J,1)
67      40      CONTINUE
68      CALL STORE2(U,V,W,H,P,I,J,K,IW,JW,IN,JN,KN,IMN,JMN,D,E,HX,HY,
69      CHI,MAR,MFH,AI,AM,AV,AP,DX,DY,CZ,DT,TAUX,TAUY,W,WR,WPH,TAI,TAM,
70      CTAV,AKT,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,
71      CITN,EX)
72      END

```

9.1.9     TMAIN5V (Main Program for Far Field)

This program simulates velocities only for unstratified shallow basins with constant vertical eddy viscosity. Steps involved in its execution are similar to those involved in the execution of TMAIN5. This program uses data element INDATA5.

## H\*TPFS.TMAIN5V

```

1      PARAMETER IN=29,JN=13,KN=6,IWN=28,JWN=12
2      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
3      CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
4      CWHLDT(IWN,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
5      CHI(IN,JN),HX(IN,JN),HY(IN,JN),PAR(IN,JN),PRH(IWN,JWN),FH(IWN,JWN),
6      COPSX(IN,JN),DPSY(IN,JN)
7      DIMENSION A3(KN)
8      DIMENSION T(I1,JN,KN),RO(IN,JN,KN),
9      CRINTX(IN,JN,KN),RINTY(IN,JN,KN)
10     CALL READ2(U,V,WH,P,I,J,K,I1,JW,IN,JN,KN,IWN,JWN,D,F,HX,HY,HI,
11     CMR,MRH,A1,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT
12     C,CB,CW,A,B,C,EUL,T,PO,TE,PREF,TREF,TO,TAMB,TOT,ITN,EX)
13     INH1=IN-1
14     READ 1, IRUN
15     READ 1, LN,LLN
16     1      FORMAT (16I5)
17     READ 1,NDTGR
18     READ 2, VVIS,ABR
19     A3(1)=VVIS
20     A3(2)=VVIS
21     A3(3)=VVIS
22     A3(4)=VVIS
23     A3(5)=VVIS
24     A3(6)=VVIS
25     B3=VVIS
26     READ 2, A1,AH,AV,AP
27     READ 2, EPS,MAXIT,OMEGA,ARBP
28     READ 2, DX,DY,DZ
29     READ 2,CC
30     READ 2,TAUXH,TALYM
31     READ 2,DT
32     READ 2, TAI,TAH,TAV
33     READ 2, A,B,C
34     READ 2, TO
35     READ 2, AKT,EUL,CW,CB
36     READ 2,TAMB,TINM
37     READ 2,IIN,JIN,IOUT,JOUT,UINH,VINH
38     2      FORMAT (I)
39     DL2=DX*DX
40     TREF=TO
41     RREF=A*B*TO+C*TC*TO
42     4      CONTINUE
43     TE=(TAMB-TREF)/TREF
44     DO 5 L=1,LN
45     TTOT=TTOT+DT
46     CALL TIMEP(TAMB,AKT,UINH,VINH,TINM,TAUXH,TAUYH,TTOT,DT)
47     CALL INLET(I,J,K,IN,JN,KN,U,V,W,H,G,TTOT,DT,
48     CNDTGR,TAUXH,TAUYH,TAUX,TAUY,TINM,T,IIN,JIN,UINH,VINH)
49     CALL OUTVEL(I,J,K,IN,JN,KN,U,V,W,H,G,PAR,IOUT,JCUT,IIN,JIN)
50     CALL ERRCP(IWN,JWN,I1,JW,DT,WH,WHLDT,KN,MRH)
51     CALL WHTCP(IWN,JWN,I1,JW,KN,WH,K,PRH)
52     CALL WHATIJ(I,J,K,I1,JW,IN,JN,KN,IWN,JWN,W,WH,PAR)
53     CALL INTE(I,J,K,IN,JN,KN,U,V,W,H,HI,HX,PY,PAR,XINT,YINT,A3,AI,
54     CAH,AV,TAUX,TAUY,DX,DY,DZ,D,E,DT,DPSX,DPSY,AP,T,TREF)
55     CALL CORINT(I,J,K,I1,JN,KN,ABR,U,V,XINT,YINT,DZ,HI,PAR)
56     CALL ROINTX(I,J,K,IN,JN,KN,LX,DY,DZ,RO,AP,EUL,HI,

```

ORIGINAL  
OF POOR QUALITY

```

57 CMAR,RINTX,HX,XIAT)
58 CALL ROINTY(I,J,K,IN,JN,KN,CX,DY,DZ,RO,AP,EUL,HI,PAR,
59 CRINTY,HY,YINT)
60 CALL DPSXY(I,J,IN,JN,IW,JW,IW,JW,CPSX,DPSY,P,DX,DY,MAR)
61 CALL FORCE(I,J,IW,JW,XINT,YINT,WH,DT,CX,DY,HI,HX,HY,MRH,
62 CDPSX,DPSY,FH,AP,IN,JN,IW,JW,RINTX,RINTY,U,V,EUL,ABR,MAR,KN)
63 CALL PREIEPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FH,DL2,CHEGA,
64 CMRH,I,J,K,IW,JW,CX,DY,EX,IW,JW,AREP)
65 CALL UVT(I,J,K,IW,JW,IN,JN,KN,IW,JW,U,V,D,E,H,G,DX,DY,DZ,
66 CRINTX,RINTY,EUL,W,DT,AI,AP,AH,AV,A3,HI,HX,HY,P,MAR,T,TREF)
67 CALL UANVC(I,J,K,IN,JN,KN,ABR,DT,U,V,H,G,HI,MAR)
68 CALL UVTCP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
69 CALL INLET(I,J,K,IN,JN,KN,U,V,H,G,TTOT,DT,
70 CNDTGR,TAUXM,TAUYM,TAUX,TAUY,TINM,T,IIN,JIN,WINM,VINM)
71 CALL OUTVEL(I,J,K,IN,JN,KN,U,V,H,G,PAR,IOUT,JOUT,IIN,JIN)
72 CALL OLDLV(I,J,K,IN,JN,KN,U,V,D,E)
73 CALL OLDUV(I,J,K,IN,JN,KN,H,G,U,V)
74 CALL RWH(I,J,K,IW,JW,IN,JN,KN,IW,JW,U,V,WH,HI,DX,DY,DZ,MRH)
75 CALL WHA TIJ(I,J,K,IW,JW,IN,JN,KN,IW,JW,W,WH,MAR)
76 5 CCNTINUE
77 CALL RWRH(I,J,K,IW,JW,IN,JN,KN,IW,JW,U,V,WH,HI,HX,HY,
78 CDX,DY,DZ,MRH,WRH)
79 CALL RWR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,HY,DZ,MAR)
80 CALL STORE2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IW,JW,D,E,HX,HY,
81 CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WPH,TAI,TAH,
82 CTAV,AKT,CB,CW,A,B,C,EUL,I,RO,TE,RREF,TREF,TO,TAMB,TTOT,
83 CITN,EX)
84 END

```



## 9.1.10 TMAIN6 (Main Program for Far Field)

This is a main program. This program prints out values stored on Unit #8. Subroutines PRPARA, TPRIN1 and PRITER print out the basic parameters. The subroutine PRPINT prints out the rigid lid pressure field. The subroutine PRUV prints out the horizontal velocity field. The subroutine PRWH prints out the vertical velocity field. The subroutine TPRIN2 prints out the temperature field. The element RTM6 provides with the computer commands necessary to execute TMAIN6 on UNIVAC -1106 computer.

## H\*TPFS.TMAINE

```

1      PARAMETER IN=29,JN=13,KN=6,IWN=28,JWN=12
2      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
3      CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
4      CWHLD(IWN,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
5      CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN),FH(IWN,JWN),
6      CDPSX(IN,JN),DPSY(IN,JN)
7      DIMENSION A3(KN),UV(JN),THETA(JN)
8      DIMENSION T(IN,JN,KN),TD(IN,JN,KN),RO(IN,JN,KN),
9      CRINTX(IN,JN,KN),RINTY(IN,JN,KN),LD(IN,JN,KN)
10     DIMENSION TW(IWN,JWN,KN),ROW(IWN,JWN,KN)
11     READ 1,IRUN
12     1    FORMAT(16I5)
13     IF (IRUN.EQ.0) GO TO 3
14     IF (IRUN.EQ.1) GO TO 16
15     IF (IRUN.EQ.2) GO TO 3
16     15   CONTINUE
17     GO TO 4
18     16   CONTINUE
19     CALL READ1(U,V,WH,P,I,J,K,IN,JN,KN,IWN,JWN,D,E,HX,HY,
20     CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TTOT)
21     GO TO 4
22     3    CONTINUE
23     CALL READ2(U,V,WH,P,I,J,K,IN,JN,KN,IWN,JWN,D,E,HX,HY,HI,
24     CHAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT
25     C,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,ITN,EX)
26     4    CONTINUE
27     CALL PRPARA(AI,AH,AV,AP,DX,DY,DZ,DT,DL2,MAXIT,EPS,OMEGA,
28     CARBP,TAUX,TAUY,TTOT,MAR,MRH,IN,JN,KN,IWN,JWN)
29     CALL TPRIN1(TAI,TAH,TAV,CB,CW,AKT,TREF,RREF,EUL,A,B,C,TE,TO)
30     CALL PRITEX(ITN,EX)
31     CALL PRPINT(IW,JW,IWN,JWN,P)
32     CALL PRUV(I,J,K,IN,JN,KN,U,V,UV,THETA,MAR)
33     CALL PRWH(IW,JW,K,IWN,JWN,KN,WRH)
34     CALL TPRIN2(I,J,K,IN,JN,KN,T,RO,TREF,MAR)
35     6    CONTINUE
36     END

```

ORIGINAL

COPY

#### 9.1.11 TMAIN4CB (Main Program for Far Field)

This program is similar to TMAIN4. This program is used to initialize velocity, temperature and pressure fields for a deep stratified cooling lake. This program reads first 27 lines of data element DATAML. Element RTM4B provides control statements to execute this program on UNIVAC-1106.

1 \*DULL(1).TMAIN4CB

```

1      PARAMETER IN=29,JN=13,KN=6,IN=28,JWN=12
2      PARAMETER NTL=14,NTLV=2
3      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),KH(14,JN,KN),
4      CWR(IN,JN,KN),KRH(IN,JN,KN),P(IN,JN),E(IN,JN,KN),
5      CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IN,JN)
6      DIMENSION A3(KN),AMINT(NTL,NTLV)
7      DIMENSION T(IN,JN,KN),RO(IN,JN,KN)
8      INMI=IN-1
9      READ 1, LN,LLN
10     1      FORMAT (14I5)
11     READ 2, VVIS,ACC
12     A3(1)=VVIS
13     A3(2)=VVIS
14     A3(3)=VVIS
15     A3(4)=VVIS
16     A3(5)=VVIS
17     A3(6)=VVIS
18     B3=VVIS
19     READ 2, AI,AH,AV,AP
20     READ 2, EPS,MAXIT,OMEGA,AREP
21     READ 2, DX,DY,DZ
22     READ 2,CC
23     READ 2,DT
24     READ 2, TAI,TAH,TAV
25     READ 2, A,B,C
26     READ 2, TO
27     READ 2, EUL,CW,CR
28     READ 2,TAMB,AKT,TAUX,TAUY
29     READ 2,CONS,AVMX,AVMN
30     READ 2,((AMINT(N,L),L=1,NTLV),N=1,NTL)
31     2      FORMAT (I)
32     DL2=DX+DX
33     TREF=TO
34     RREF=A+B*TO+C*TO*TO
35     CALL READ3(I,J,IN,JN,IW,JW,IWN,JWN,MAR,MRH)
36     CALL HITEA(I,J,K,IN,JN,KN,HI,HX,HY,CC)
37     CALL INITIAC(IN,JN,KN,IWN,JWN,U,V,W,KH,D,E,P,I,J,K,IW,JW,ARBP)
38     CALL INITD(I,J,K,IN,JN,KN,IW,JW,IWN,JWN,A,B,C,T,RO,MAR,MRH,TREF,
39     CRREF,TO,AMINT,HI,NTL,NTLV)
40     CALL GRADSI(IN,JN,KN,IWN,JWN,HI,HX,HY,MAR,MRH,DX,LY)
41     TTOT=J.0
42     ITN=0
43     EX=0.
44     GO TO 4
45     10     CONTINUE
46     3      CONTINUE
47     4      CONTINUE
48     CALL STOR3(U,V,KH,P,I,J,K,IN,JN,KN,IWN,JWN,D,E,HX,HY,
49     CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,KH,TAI,TAH,
50     CTAV,AKT,CR,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,
51     CITN,EX)
52     CALL TPRIN(I,J,K,IN,JN,KN,T,RO,TREF,MAR)
53     6      CONTINUE
54     END

```

#### 9.1.12 TMAIN4TB (Main Program for Far Field)

This program is similar to TMAIN4T. This program specifies measured initial temperatures for a deep stratified cooling lake. This program reads data element ITLK1. Element RTM4TB provides control statements to execute this program on UNIVAC-1106.

(=DULL(1), TMAIN4TD

```

1      PARAMETER IN=29, JN=13, KN=6, IIN=26, JNN=12
2      PARAMETER NTL=14, NTLV=L
3      DIMENSION U(IJ, JN, KN), V(IJ, JN, KN), W(IJ, JN, KN), WH(IIN, JNN, KN),
4      CWR(IJ, JN, KN), WRH(IIN, JNN, KN), P(IIN, JNN), D(IJ, JN, KN), E(IJ, JN, KN),
5      CHI(IJ, JN), HX(IJ, JN), HY(IJ, JN), MAR(IJ, JN), MRH(IIN, JNN)
6      DIMENSION AMINT(NTL, NTLV)
7      DIMENSION T(IJ, JN, KN), RO(IJ, JN, KN)
8      3      CONTINUE
9      CALL READCB(U, V, W, P, I, J, K, IJ, JW, IN, JN, KN, IIN, JNN, D, E, HX, HY, HI,
10     CHAR, MRH, AI, AH, AV, AP, DX, DY, DZ, DT, TAUX, TAUY, W, WR, WHH, TAI, TAH, TAV, AKT
11     C, CB, CW, A, B, C, EUL, T, RO, TE, RREF, TREF, TO, TAMB, TTOT, ITH, EX)
12     4      CONTINUE
13     READ 2, ((AMINT(N, L), L=1, NTLV), N=1, NTL)
14     2      FORMAT (1)
15     CALL INITB(I, J, K, IJ, JN, KN, IJ, JW, IIN, JNN, A, B, C, T, RO, MAR, MRH, TREF,
16     CRREF, TO, AMINT, HI, NTL, NTLV)
17     CALL INITMB(I, J, K, IJ, JN, KN, IJ, JW, IIN, JNN, A, B, C, T, RO, MAR, MRH, TREF,
18     CRREF, TO, HI)
19     TTOT=0.000
20     CALL STORCB(U, V, W, P, I, J, K, IJ, JW, IN, JN, KN, IIN, JNN, D, E, HX, HY,
21     CHI, MAR, MRH, AI, AH, AV, AP, DX, DY, DZ, DT, TAUX, TAUY, W, WR, WHH, TAI, TAH,
22     CTAV, AKT, CB, CW, A, B, C, EUL, T, RO, TE, RREF, TREF, TO, TAMB, TTOT,
23     CITN, EX)
24     CALL TPRIN9(I, J, K, IJ, JN, KN, T, RO, TREF, MAR)
25     6      CONTINUE
26     END

```

### 9.1.13 TMAIN5B (Main Program for Far Field)

This program is similar to TMAIN5. This program is used to simulate velocities and temperatures in a coupled fashion for a deep stratified cooling lake. This program reads data element DATAML5. Element RTM5B provides control statements to execute this program on UNIVAC-1106.

4\*DULL(1).TMAIN=8

```

1      PARAMETER IN=29, JN=13, KN=6, IWN=28, JWN=12
2      DIMENSION U(IN, JN, KN), V(IN, JN, KN), W(IN, JN, KN), WH(IN, JN, KN),
3      CWR(IN, JN, KN), WRH(IN, JN, KN), F(IN, JN), D(IN, JN, KN), E(IN, JN, KN),
4      WHLOT(IW, JWN), XINT(IN, JN), YINT(IN, JN), H(IN, JN, KN), C(IN, JN, KN),
5      CHI(IN, JN), HX(IN, JN), HY(IN, JN), MAR(IN, JN), MRH(IWN, JWN), FH(IWN, JWN),
6      CPSX(IN, JN), DPSY(IN, JN)
7      DIMENSION A3(KN)
8      DIMENSION T(IN, JN, KN), TD(IN, JN, KN), RO(IN, JN, KN),
9      CRINTX(IN, JN, KN), RINTY(IN, JN, KN), WD(IN, JN, KN)
10     CALL READ23(U, V, WH, P, I, J, K, IW, JW, IN, JN, KN, IWN, JWN, D, E, HX, HY, HI,
11     CHAR, MRH, AI, AH, AV, AP, DX, DY, DZ, DT, TAUX, TAUY, A, B, C, WRH, TAI, TAH, TAV, AKT,
12     C, CB, CW, A, B, C, EUL, T, RO, TE, RREF, TREF, TO, TAMB, TTOT, ITN, EX)
13     INM=IN-1
14     READ 1, LN, LLN
15     1   FORMAT (16I5)
16     READ 2, VVIS, ABP
17     A3(1)=VVIS
18     A3(2)=VVIS
19     A3(3)=VVIS
20     A3(4)=VVIS
21     A3(5)=VVIS
22     A3(6)=VVIS
23     B3=VVIS
24     READ 2, AI, AH, AV, AP
25     READ 2, EPS, MAXIT, OMEGA, ARBP
26     READ 2, DX, DY, DZ
27     READ 2, CC
28     READ 2, DT
29     READ 2, TAI, TAH, TAV
30     READ 2, A, B, C
31     READ 2, TO
32     READ 2, EUL, CW, CB
33     READ 2, TAMB, AKT, TAUX, TAUY
34     READ 2, CONS, AVMX, AVMN
35     2   FORMAT ( )
36     DLZ=DX+DX
37     TREF=TO
38     RREF=A+B*TO+C*TO*TO
39     4   CONTINUE
40     TE=(TAMB-TREF)/TREF
41     CALL INLETS(I, J, K, IN, JN, KN, U, V, H, G, T)
42     DO 5 L=1, LN
43     CALL ERROR(IWN, JWN, IW, JW, DT, WH, WHLOT, KN, MRH)
44     CALL WHTOP(IW, JW, IWN, JWN, KN, WH, K, MRH)
45     CALL WHATIJ(I, J, K, IW, JW, IN, JN, KN, IWN, JWN, W, WH, MAR)
46     CALL INTEB(I, J, K, IN, JN, KN, U, V, W, HI, HX, HY, MAR, XINT, YINT, A3, AI,
47     CAH, AV, TAUX, TAUY, DX, DY, DZ, D, E, DT, CPSX, DPSY, AP, T, TREF
48     C, CONS, AVMX, AVMN)
49     CALL CORINT(I, J, K, IN, JN, KN, ABR, U, V, XINT, YINT, DZ, HI, MAR)
50     CALL RCINTX(I, J, K, IN, JN, KN, DX, DY, DZ, RO, AP, EUL, HI,
51     CHAR, RINTX, HX, XINT)
52     CALL RCINTY(I, J, K, IN, JN, KN, DX, DY, DZ, RC, AP, EUL, HI, MAR,
53     CRINTY, HY, YINT)
54     CALL DPSXY(I, J, IN, JN, IW, JW, IWN, JWN, CPSX, DPSY, P, DX, DY, MAR)
55     CALL FORCE(I, J, IW, JW, XINT, YINT, WHLOT, IX, DY, HI, HX, HY, MRH,
56     CPSX, DPSY, FH, AP, IN, JN, IWN, JWN, RINTX, RINTY, U, V, EUL, ABR, MAR, KN)

```



```

57 CALL PRE1CEPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FH,DL2,OMEGA,
58 CMRH,I,J,K,IN,JW,DX,DY,EX,IN,JW,ARLF)
59 CALL UVTI(I,J,K,IN,JW,IN,JN,KN,IN,JW,U,V,D,E,H,G,DX,DY,DZ,
60 CRIP,IX,KINTY,EUL,W,DT,AI,AP,AH,AV,A3,HI,HX,HY,P,MAR,T,TREF
61 C,CONS,AVMX,AVMN)
62 CALL UANVC(I,J,K,IN,JN,KN,AGR,DT,U,V,H,G,HI,MAR)
63 CALL UVTOPH,C,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
64 CALL OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
65 CALL OLDUV(I,J,K,IN,JN,KN,H,G,U,V)
66 CALL RWH(I,J,K,IN,JN,IN,JN,KN,ZNA,JW,U,V,WH,HI,DX,DY,DZ,MRH)
67 CALL WHATTJ(I,J,K,IN,JN,IN,JN,KN,ZNA,JW,U,V,WH,MAR)
68 DO 20 I=1,IN
69 DO 20 J=1,JN
70 WD(I,J,1)=W(I,J,1)
71 20 CONTINUE
72 DO 30 I=1,IN
73 DO 30 J=1,JN
74 W(I,J,1)=D.D
75 30 CONTINUE
76 CALL TEMI43(I,J,K,IN,JN,KN,U,V,T,TD,DX,CB,DY,DZ,W,DT,TAI,TAH,TAV,
77 CB3,HI,HX,HY,MAR,AKT,TREF,TAMB,A3,CONS,AVMX,AVMN)
78 CALL TEMS2B(I,J,K,IN,JN,KN,TD,DX,DY,DZ,MAR,CB,HI,AKT,CW,TAMB,HX,
79 CHY,T,TREF,TAV,TAI,TAH,D3,DT,A3,CONS,AVMX,AVMN)
80 CALL OLD(T,I,J,K,IN,JN,KN,TD,T)
81 CALL TSTAP(I,J,K,IN,JN,KN,T,MAR)
82 CALL DENSTB(I,J,K,IN,JW,IN,JN,KN,IN,JN,A,B,C,MAR,MRH,T,
83 CRO,RREF,TREF)
84 DO 40 I=1,IN
85 DO 40 J=1,JN
86 W(I,J,1)=WD(I,J,1)
87 40 CONTINUE
88 5 CONTINUE
89 CALL RWRH(I,J,K,IN,JW,IN,JN,KN,IN,JN,U,V,WH,HI,HX,HY,
90 CDX,DY,DZ,MRH,VRH)
91 CALL RWR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,HY,DZ,MAR)
92 CALL STOREB(U,V,WH,P,I,J,K,IN,JW,IN,JN,KN,IN,JW,C,E,HX,HY,
93 CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WA,WRH,TAI,TAH,
94 CTAV,AKT,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,
95 CITN,EX)
96 END

```

#### 9.1.14 TMAIN5VB (Main Program for Far Field)

This program is similar to TMAIN5V. This program is used to simulate velocities only for a deep stratified cooling lake. This program uses data element DATAML5. Element RTM5VB provides control statements to execute this program on UNIVAC-1106.

1=DULL(1),THAINSVB

```

1      PARAMETER IN=27, JN=13, KN=6, IKN=28, JKN=12
2      DIMENSION UI(I,N,KN),V(I,N,KN),W(I,N,KN),WH(IWN,JWN,KN),
3      CWR(I,N,KN),WHM(IW,JW,KN),P(IW,JW),D(IW,JW,KN),E(I,N,KN),
4      C=HLOT(IW,JW),XINT(I,N,KN),YINT(I,N,KN),H(I,N,KN),G(I,N,KN),
5      CH(I,N,KN),HX(I,N,KN),HY(I,N,KN),MAR(I,N,KN),MRH(IW,JW),FH(IW,JW),
6      CPSX(I,N,KN),CPSY(I,N,KN)
7      DIMENSION A3(KN)
8      DIMENSION T(I,N,KN),RO(I,N,KN),
9      CRINTX(I,N,KN),CRINTY(I,N,KN)
10     CALL READC(U,V,W,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HY,HI,
11     CHAR,MRH,A3,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,N,NR,WRH,TAI,TAH,TAV,AKT
12     C,CC,CW,A,B,C,EUL,1,FO,TE,PREF,TREF,TC,TAMB,TTOT,ITN,EX)
13     IKN1=IKN-1
14     READ 1, LK,LLN
15     1      FORMAT (1E15)
16     READ 2, VVIS,ABF
17     A3(1)=VVIS
18     A3(2)=VVIS
19     A3(3)=VVIS
20     A3(4)=VVIS
21     A3(5)=VVIS
22     A3(6)=VVIS
23     B3=VVIS
24     READ 2, AI,AH,AV,AP
25     READ 2, EPS,MAXIT,OMEGA,AFBP
26     READ 2, DX,DY,DZ
27     READ 2,CC
28     READ 2,DT
29     READ 2, TAI,TAH,TAV
30     READ 2, A,B,C
31     READ 2, TO
32     READ 2, EUL,CW,CB
33     READ 2,TAMB,AKT,TAUX,TAUY
34     READ 2,CONS,AVMX,AVMY
35     2      FORMAT (1)
36     DL2=DX*DX
37     TREF=TO
38     RREF=A+B*TO+C*TC+TO
39     4      CONTINUE
40     TE=(TAMB-TREF)/TREF
41     CALL INLET3(I,J,K,IN,JN,KN,U,V,W,G,T)
42     DO 5 L=1,LN
43     TTOT=TTOT+DT
44     CALL ERROR(IW,JW,IW,JW,DT,WH,WHLOT,KN,MRH)
45     CALL WHTCF(IW,JW,IW,JW,KN,WH,K,MRH)
46     CALL WHATEU(I,J,K,IN,JN,IN,JN,KN,IW,JW,W,WH,MAR)
47     CALL INTER(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3,A1,
48     CAH,AV,TAUX,TAUY,DZ,DY,DZ,D,E,DT,CPSX,CPSY,AP,T,TREF
49     C,CONS,AVMX,AVMY)
50     CALL CORINT(I,J,K,IN,JN,KN,ABR,U,V,XINT,YINT,DZ,HI,MAR)
51     CALL POINTAI(I,J,K,IN,JN,KN,DX,DY,DZ,AC,AP,EUL,HI,
52     CHAR,CRINTX,HX,YINT)
53     CALL POINTY(I,J,K,IN,JN,KN,DX,DY,DZ,AC,AP,EUL,IT,MAR,
54     CRINTY,HY,YINT)
55     CALL DPSXY(I,J,IN,JN,IW,JW,IW,JW,CPSX,CPSY,P,DX,DY,MAR)
56     CALL FORCE(I,J,IW,JW,XINT,YINT,WHLOT,DX,DY,HI,HX,IY,MRH,

```

57 CDPSX,DPSY,FH,AP,IN,JN,IWN,JWN,RINTX,RINTY,U,V,EUL,APR,MAR,KN)  
 58 CALL PRE11EPS,MAXIT,IN,JN,P,ITN,CDPSX,DPSY,FH,DL2,OMEGA,  
 59 CMRH,I,J,K,IW,JW,DX,DY,EX,IW,JWN,ARBP)  
 60 CALL UVTR(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,D,E,H,G,DX,DY,DZ,  
 61 CRINTX,RINTY,LUL,DT,AI,AP,AH,AV,A3,HI,HX,IW,P,MAR,T,TREF  
 62 C,CONS,AVHX,AVMN)  
 63 CALL UANVC(I,J,K,IN,JN,KN,APR,DT,U,V,H,G,HI,MAR)  
 64 CALL UVTOP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)  
 65 CALL OLDUV(I,J,K,IN,JN,KN,U,V,D,E)  
 66 CALL OLDUV(I,J,K,IN,JN,KN,H,G,U,V)  
 67 CALL RWH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,DX,DY,DZ,MRH)  
 68 CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)  
 69 S CONTINUE  
 70 CALL RWRH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,HX,HY,  
 71 CDX,DY,DZ,MRH,WRH)  
 72 CALL RWS(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,HY,DZ,MAR)  
 73 CALL STOR23(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,  
 74 CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,  
 75 CTAV,AKT,CB,CW,A,B,C,EUL,T,RC,TE,RREF,TREF,TO,TAMB,TTOT,  
 76 CITN,EX)  
 77 END

#### 9.1.15 TMAIN5TB (Main Program for Far Field)

This program is similar to TMAIN5T. This program is used to simulate temperatures only for a deep stratified cooling lake. This program uses data element DATAML5. Element RTM5B provides with the control statements needed to execute this program on UNIVAC-1106.

M=DULL(1).TPAINSTB

```

1      PARAMETER IN=29,JN=17,KN=6,IN=28,JWN=12
2      DIMENSION U(IN,JN,KN),V(IN,JN,KN),K(IN,JN,KN),KH(IN,JN,KN),
3      CWR(IN,JN,KN),MRH(IN,JN,KN),P(IN,JN,KN),D(IN,JN,KN),L(IN,JN,KN),
4      CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IN,JN)
5      DIMENSION C(IN,JN,KN),H(IN,JN,KN)
6      DIMENSION A3(KN)
7      DIMENSION T(IN,JN,KN),TD(IN,JN,KN),RO(IN,JN,KN),LD(IN,JN,KN)
8      CALL READ2B(U,V,AH,P,I,J,K,TL,JW,IN,JN,KN,IWN,JWN,D,Z,HX,HY,HI,
9      CHAR,MRH,AI,AH,AV,AP,DX,DY,CZ,DT,TAUX,TAUY,W,AR,WRH,TAI,TAH,TAV,AKT
10     C,CB,CH,A,B,C,EUL,T,PC,TE,RREF,TREF,TO,TAMB,TTOT,ITH,EX)
11     INMI=IN-1
12     READ 1, LN,LLN
13     1    FORMAT (16I5)
14     READ 2, VVIS,ABR
15     A3(1)=VVIS
16     A3(2)=VVIS
17     A3(3)=VVIS
18     A3(4)=VVIS
19     A3(5)=VVIS
20     A3(6)=VVIS
21     B3=VVIS
22     READ 2, AI,AH,AV,AP
23     READ 2, EPS,MAXIT,OMEGA,AREP
24     READ 2, DX,DY,DZ
25     READ 2,CC
26     READ 2,DT
27     READ 2, TAI,TAH,TAV
28     READ 2, A,B,C
29     READ 2, TO
30     READ 2, EUL,CH,CB
31     READ 2,TAMB,AKT,TAUX,TAUY
32     READ 2,CONS,AVMX,AVMH
33     2    FORMAT (1
34     DL2=DX*DX
35     TREF=TO
36     RREF=A*B*TO+C*TO*TO
37     4    CONTINUE
38     DO 20 I=1,IN
39     DO 20 J=1,JN
40     WD(I,J,1)=W(I,J,1)
41     20    CONTINUE
42     DO 30 I=1,IN
43     DO 30 J=1,JN
44     W(I,J,1)=0.0
45     30    CONTINUE
46     CALL INLETS(I,J,K,IN,JN,KN,U,V,H,G,T)
47     DO 6 LL=1,LLN
48     ITOT=1TOT+DT
49     CALL TCM148(I,J,K,IN,JN,KN,U,V,T,TD,DX,CB,DY,DZ,W,DT,TAI,TAH,TAV,
50     CB3,HI,HX,HY,MAR,AKT,TREF,TAMB,A3,CONS,AVMX,AVMH)
51     CALL TEM2CB(I,J,K,IN,JN,KN,TE,DX,DY,DZ,MAR,CE,HI,AKT,CB,TAMB,HX,
52     CHY,T,TREF,TAV,TAI,TAH,L3,(T,A3,CONS,AVMX,AVMH)
53     CALL OLOT(I,J,K,IN,JN,KN,TD,T)
54     CALL TSTAB(I,J,K,IN,JN,KN,T,MAR)
55     6    CONTINUE
56     CALL DENSTB(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,A,B,C,MAR,MRH,T,

```

```

57      CRO,RREF,TREF)
58      DO 40 I=1,IN
59      DO 40 J=1,JN
60      W(I,J,1)=WD(I,J,1)
61      40 CONTINUE
62      CALL STOP2B(U,V,W,H,P,I,J,K,IN,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,
63      CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WPH,TAI,TAH,
64      CTAV,AKT,CB,CW,A,B,C,EUL,T,RC,TE,RREF,TREF,TO,TAMB,TTOT,
65      CITN,EX)
66      END

```

#### 9.1.16 TMAIN6B (Main Program for Far Field)

This program is similar to TMAIN6. This program prints out the results for the simulation of velocity and/or temperature in a deep stratified cooling pond. Element RTM6B provides with the control statements necessary to execute this program on UNIVAC-1106.



MODJLL(1),THAIN68

```

1      PARAMETER IN=29,JN=13,KN=6,IN=28,JWN=12
2      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),LH(IN,JN,KN),
3      CWR(IN,JN,KN),LKH(IN,JN,KN),P(IN,JN,KN),D(IN,JN,KN),L(IN,JN,KN),
4      CHI(IN,JN),HX(1,JN),HY(1,JN),MAR(IN,JN),MCH(IN,JN)
5      DIMENSION A(KN),UV(JN),THETA(JN)
6      DIMENSION T(IN,JN,KN),RO(IN,JN,KN)
7      INM1=IN-1
8      READ 1, LN,LLN
9      1  FORMAT (16I5)
10     READ 2, VVIS,ABR
11     A3(1)=VVIS
12     A3(2)=VVIS
13     A3(3)=VVIS
14     A3(4)=VVIS
15     A3(5)=VVIS
16     A3(6)=VVIS
17     B3=VVIS
18     READ 2, AI,AH,AV,AP
19     READ 2, EPS,MAXIT,OMEGA,ARBP
20     READ 2, DX,DY,DZ
21     READ 2,CC
22     READ 2,DT
23     READ 2, TAI,TAH,TAV
24     READ 2, A,B,C
25     READ 2, TC
26     READ 2, EUL,CW,CB
27     READ 2,TAMB,AKT,TAUX,TAUY
28     READ 2,CONS,AVMX,AVMN
29     2  FORMAT ( )
30     DL2=DX*DX
31     CALL READDB(U,V,WH,P,I,J,K,IN,JW,IN,JN,KN,IWA,JWN,D,E,HX,HY,HI,
32     CHAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,U,WR,WHF,TAI,TAH,TAV,AKT
33     C,CB,CW,A,B,C,EUL,T,RO,TC,RREF,TREF,TO,TAMB,TTOT,TH,EX)
34     4  CONTINUE
35     CALL PRPAPA(AI,AH,AV,AP,DX,DY,DZ,DT,DL2,MAXIT,EPS,OMEGA,
36     CARBP,TAUX,TAUY,TTOT,MAR,MRH,IN,JN,IKN,JWN)
37     CALL TPRINI(TAI,TAH,TAV,CB,CW,AKT,TREF,RREF,EUL,A,B,C,TC,TO)
38     CALL PRITEX(ITN,EX)
39     CALL PRPINT(IN,JN,IKN,JWN,F)
40     CALL PRUVA(I,J,K,IN,JN,KN,U,V,UV,THETA,MAR)
41     CALL PRWH(IW,JN,K,IKN,JN,KN,WRH)
42     CALL TPRINAI(I,J,K,IN,JN,KN,T,RO,TREF,MAR)
43     6  CONTINUE
44     END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

SUBROUTINES FOR NEAR AND FAR-FIELD

1111  
INTERCOMPARISON

## 9.2 Subroutines for the Near Field and Far Field

The subroutines for the Near field and Far field are given in alphabetical order in this section. These subroutines are called by the main programs. The Fortran symbol explanation can be seen in section 3 where they are explained in alphabetical order.

### 9.2.1 CORINT

This program adds integral of Coriolis component to XINT and YINT. XINT and YINT are calculated in the subroutine INTE. XINT and YINT are integral of x and y components on the right hand side of Poisson's equation (Eq 2.17, Vol.1)

```

1      SUBROUTINE CORINT(I,J,K,IN,JN,KN,ABR,U,V,XINT,YINT,DZ,HI,MAR)
2      DIMENSION U(IN,JN,KN),V(IN,JN,KN),XINT(IN,JN),YINT(IN,JN),HI(IN,
3      CUN),PAR(IN,JN)
4      DO 10 I=1,IN
5      DO 10 J=1,JN
6      IF (MAR(I,J).LT.11) GO TO 9
7      DO 8 K=2,KN
8      XINT(I,J)=XINT(I,J)-ABR*HI(I,J)*(V(I,J,K-1)+V(I,J,K))*DZ/2
9      YINT(I,J)=YINT(I,J)+ABR*HI(I,J)*(U(I,J,K-1)+U(I,J,K))*DZ/2
10     CONTINUE
11     9 CONTINUE
12     10 CONTINUE
13     RETURN
14     END

```

### 9.2.2 CURNT

This program puts current into the model. This program must be changed depending on the direction and magnitude of the current.

(This subroutine is not used in the sample case, as the current is not considered)

\*00C.CURNTS

```

1      SUBROUTINE CURNTS(I,J,K,IN,JN,KN,U,V,D,E,H,G)
2      DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),
3      CE(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN)
4      KKM1=KN-1
5      DO 10 I=1,IN
6      DO 10 K=1,KKM1
7      DO 10 J=2,JN
8      U(I,J,K)=-0.1
9      V(I,J,K)=0.0
10     D(I,J,K)=-0.1
11     E(I,J,K)=0.0
12     H(I,J,K)=-0.1
13     G(I,J,K)=0.0
14     10 CONTINUE
15     RETURN
16     END

```

### 9.2.3 CWXY

This program computes horizontal temperature gradients at the vertical walls in  $\alpha$  and  $\beta$  directions from the heat flux in  $x$  and  $y$  directions.



I=DOC.CWXY

```

1  SUBROUTINE CWXY(CWX,CWY,CW,I,J,K,IN,JN,KN,HI,HX,HY,T,DZ)
2  DIMENSION T(IN,JN,KN),HI(IN,JN),HX(IN,JN),HY(IN,JN)
3  IF (K.EQ.1) GO TO 99
4  IF (K.EQ.KN) GO TO 101
5  D1T2=(T(I,J,K+1)-T(I,J,K-1))/(2*DZ)
6  GO TO 206
7  99  CONTINUE
8  D1T2=(4*T(I,J,K+1)-3*T(I,J,K)-T(I,J,K+2))/(2*DZ)
9  GO TO 206
10 101 CONTINUE
11 D1T2=(3*T(I,J,K)+T(I,J,K-2)-4*T(I,J,K-1))/(2*DZ)
12 206 CONTINUE
13 CWX=CW+(K-1)*DZ*HX(I,J)+D1T2/HI(I,J)
14 CWY=CW+(K-1)*DZ*HY(I,J)+D1T2/HI(I,J)
15 RETURN
16 END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

#### 9.2.4 DENSTA

This subroutine is used for the far field unstratified cooling reservoir. The subroutine is similar to the subroutine DENSTY. The only difference being that the matrix ROW (densities at the half grid points) is eliminated to save computer core space.

SKH=DULL(1),DENSTA

```

1 C*****
2 C      THE FOLLOWING PROGRAM CALCULATES THE DENSITY FIELD FROM
3 C      THE TEMPERATURE FIELD
4 C*****
5 SUBROUTINE DENSTA(I,J,K,IN,JN,KN,INN,JWN,A,B,C,
6   CHAR,HRH,
7   CT,TW,RJ,RREF,TREF)
8   DIMENSION RO(IN,JN,KN),T(IN,JN,KN)
9   DIMENSION TW(INN,JWN,KN)
10  DIMENSION MAR(IN,JN),HRH(INN,JWN)
11  DO 10 I=1,IN
12  DO 10 J=1,JN
13  IF (MAR(I,J).EQ.0) GO TO 12
14  DO 11 K=1,KN
15  TEM=T(I,J,K)*TREF+TREF
16  RA=A+B*TEM+C*TEM*TEM
17  RO(I,J,K)=(R-RREF)/RREF
18 11 CONTINUE
19 12 CONTINUE
20 10 CONTINUE
21 RETURN
22 END

```

### 9.2.5 DENSTB

This subroutine is used for the far field stratified cooling lake. The subroutine is similar to DENSTY, the only difference being that the matrices TW (temperatures at the half grid points) and ROW (densities at the half grid points) are eliminated to save computer core space.

SKM\*OULL(1),DENST6

```

1 C*****
2 C      THE FOLLOWING PROGRAM CALCULATES THE DENSITY FIELD FROM
3 C      THE TEMPERATURE FIELD
4 C*****
5      SUBROUTINE DENST6(I,J,K,IW,JW,IN,JN,KN,INN,JWN,A,B,C,
6      CHAR,MRH,
7      CT,RO,RREF,TREF)
8      DIMENSION RO(IN,JN,KN),T(IN,JN,KN)
9      DIMENSION MAR(IN,JN),MRH(IWN,JWN)
10     DO 10 I=1,IN
11     DO 10 J=1,JN
12     IF (MAR(I,J).EQ.0) GO TO 12
13     DO 11 K=1,KN
14     TEM=T(I,J,K)*TREF+TREF
15     R=A+B*TEM+C*TEM*TEM
16     RO(I,J,K)=(R-RREF)/RREF
17     11 CONTINUE
18     12 CONTINUE
19     10 CONTINUE
20     RETURN
21     END

```

## 9.2.6 DENSTY

This program uses the equation of state and computes density field from the temperature field

Eq of State

$$\rho = A + B(T) + C(T)^2$$

In the program

$$\rho = A + B(\text{Tem}) + C(\text{TEM})^2$$

Where A,B,C are constants and there values are A = 1029431,

B = -0.00002 and C = -0.0000048

## \*DOC.DENSTY

```

1 C*****
2 C      THE FOLLOWING PROGRAM CALCULATES THE DENSITY FIELD FROM
3 C      THE TEMPERATURE FIELD
4 C*****
5     SUBROUTINE DENSTY(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,A,B,C,
6     CHAR,MRH,
7     CT,TW,RO,ROW,RREF,TREF)
8     DIMENSION RO(IN,JN,KN),T(IN,JN,KN)
9     DIMENSION ROW(IWN,JWN,KN),TW(IWN,JWN,KN)
10    DIMENSION MAR(IN,JN),MRH(IWN,JWN)
11    DO 10 I=1,IN
12    DO 10 J=1,JN
13    IF (MAR(I,J).EQ.0) GO TO 12
14    DO 11 K=1,KN
15    TEM=T(I,J,K)*TREF+TREF
16    R=A+B*TEM+C*TEM*TEM
17    RO(I,J,K)=(R-RREF)/RREF
18    11 CONTINUE
19    12 CONTINUE
20    10 CONTINUE
21    DO 20 IW=1,IWN
22    DO 20 JW=1,JWN
23    IF (MRH(IW,JW).EQ.0) GO TO 22
24    DO 21 K=1,KN
25    TEMW=TW(IW,JW,K)*TREF+TREF
26    RW=A+B*TEMW+C*TEMW*TEMW
27    ROW(IW,JW,K)=(RW-RREF)/RREF
28    21 CONTINUE
29    22 CONTINUE
30    20 CONTINUE
31    RETURN
32    END

```

### 9.2.7 DINERU

This subroutine computes DIHUUX, DIHUVX. This program is called in INTE. The results are used in Poisson equation for pressure.



\*DOC.DINERU

```

1  SUBROUTINE DINERU(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1HUUX,D1HUVX,MAR)
2  DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),MAR(IN,JN)
3  IF(MAR(I,J).EQ.C) GO TO 50
4  IF(MAR(I,J).EQ.1) GO TO 31
5  IF(MAR(I,J).EQ.2) GO TO 32
6  IF(MAR(I,J).EQ.3) GO TO 33
7  IF(MAR(I,J).EQ.4) GO TO 34
8  IF(MAR(I,J).EQ.5) GO TO 35
9  IF(MAR(I,J).EQ.6) GO TO 36
10 IF(MAR(I,J).EQ.7) GO TO 37
11 IF(MAR(I,J).EQ.8) GO TO 38
12 IF(MAR(I,J).EQ.9) GO TO 39
13 IF(MAR(I,J).EQ.10) GO TO 40
14 D1HUUX=(U(I+1,J,K)*U(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
15 C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
16 D1HUVX=(U(I+1,J,K)*V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
17 C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
18 GO TO 50
19 31 CONTINUE
20 D1HUUX=(U(I+1,J,K)*U(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
21 C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
22 D1HUVX=(U(I+1,J,K)*V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
23 C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
24 GO TO 50
25 32 CONTINUE
26 D1HUUX=(U(I+1,J,K)*U(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
27 C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
28 D1HUVX=(U(I+1,J,K)*V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
29 C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
30 GO TO 50
31 33 CONTINUE
32 D1HUUX=(4*HI(I+1,J)*U(I+1,J,K)*U(I+1,J,K)-3*HI(I,J)*U(I,J,K)
33 C*U(I,J,K)-HI(I+2,J)*U(I+2,J,K)*U(I+2,J,K))/(2*DX)
34 D1HUVX=(4*HI(I+1,J)*U(I+1,J,K)*V(I+1,J,K)-3*HI(I,J)*U(I,J,K)
35 C*V(I,J,K)-HI(I+2,J)*U(I+2,J,K)*V(I+2,J,K))/(2*DX)
36 GO TO 50
37 34 CONTINUE
38 D1HUUX=(3*HI(I,J)*U(I,J,K)*U(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
39 C*U(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)*U(I-2,J,K))/(2*DX)
40 D1HUVX=(3*HI(I,J)*U(I,J,K)*V(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
41 C*V(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)*V(I-2,J,K))/(2*DX)
42 GO TO 50
43 35 CONTINUE
44 D1HUUX=(4*HI(I+1,J)*U(I+1,J,K)*U(I+1,J,K)-3*HI(I,J)*U(I,J,K)
45 C*U(I,J,K)-HI(I+2,J)*U(I+2,J,K)*U(I+2,J,K))/(2*DX)
46 D1HUVX=(4*HI(I+1,J)*U(I+1,J,K)*V(I+1,J,K)-3*HI(I,J)*U(I,J,K)
47 C*V(I,J,K)-HI(I+2,J)*U(I+2,J,K)*V(I+2,J,K))/(2*DX)
48 GO TO 50
49 36 CONTINUE
50 D1HUUX=(U(I+1,J,K)*U(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
51 C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
52 D1HUVX=(U(I+1,J,K)*V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
53 C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
54 GO TO 50
55 37 CONTINUE
56 D1HUUX=(4*HI(I+1,J)*U(I+1,J,K)*U(I+1,J,K)-3*HI(I,J)*U(I,J,K)

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

57      C*U(I,J,K)-HI(I+2,J)*U(I+2,J,K)+U(I+2,J,K))/(2*DX)
58      D1HUVX=(4*HI(I+1,J)*U(I+1,J,K)+V(I+1,J,K)-3*HI(I,J)*U(I,J,K)
59      C*V(I,J,K)-HI(I+2,J)*U(I+2,J,K)+V(I+2,J,K))/(2*DX)
60      GO TO 50
61      38      CONTINUE
62      D1HUUX=(U(I+1,J,K)+U(I+1,J,K)+HI(I+1,J)-U(I-1,J,K)
63      C*U(I-1,J,K)+HI(I-1,J))/(2*DX)
64      D1HUVX=(U(I+1,J,K)+V(I+1,J,K)+HI(I+1,J)-U(I-1,J,K)
65      C*V(I-1,J,K)+HI(I-1,J))/(2*DX)
66      GO TO 50
67      39      CONTINUE
68      D1HUUX=(3*HI(I,J)+U(I,J,K)+U(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
69      C*U(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)+U(I-2,J,K))/(2*DX)
70      D1HUVX=(3*HI(I,J)+U(I,J,K)+V(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
71      C*V(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)+V(I-2,J,K))/(2*DX)
72      GO TO 50
73      40      CONTINUE
74      D1HUUX=(3*HI(I,J)+U(I,J,K)+U(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
75      C*U(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)+U(I-2,J,K))/(2*DX)
76      D1HUVX=(3*HI(I,J)+U(I,J,K)+V(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
77      C*V(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)+V(I-2,J,K))/(2*DX)
78      50      CONTINUE
79      RETURN
80      END

```

## 9.2.8 DUVY

This program computes DIHUVY. The program is called in by INTE,  $\frac{\partial}{\partial \beta}$  (huv) is computed for interior, boundary and corner points by the scheme similar to the one used in DINERU.

1000C.DUVY

```

1  SUBROUTINE DUVY(I,J,K,IN,JN,KN,U,V,HI,DY,DIHUVY,HAR)
2  DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),HAR(IN,JN)
3  IF(HAR(I,J).EQ.C) GO TO 50
4  IF(HAR(I,J).EQ.1) GO TO 31
5  IF(HAR(I,J).EQ.2) GO TO 32
6  IF(HAR(I,J).EQ.3) GO TO 33
7  IF(HAR(I,J).EQ.4) GO TO 34
8  IF(HAR(I,J).EQ.5) GO TO 35
9  IF(HAR(I,J).EQ.6) GO TO 36
10 IF(HAR(I,J).EQ.7) GO TO 37
11 IF(HAR(I,J).EQ.8) GO TO 38
12 IF(HAR(I,J).EQ.9) GO TO 39
13 IF(HAR(I,J).EQ.10) GO TO 40
14 DIHUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
15 C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
16 GO TO 50
17 31 CONTINUE
18 DIHUVY=(3*HI(I,J)*U(I,J,K)*V(I,J,K)-4*HI(I,J-1)*U(I,J-1,K)
19 C*V(I,J-1,K)*HI(I,J-2)+U(I,J-2,K)*V(I,J-2,K))/(2*DY)
20 GO TO 50
21 32 CONTINUE
22 DIHUVY=(4*HI(I,J+1)*U(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*
23 CU(I,J,K)*V(I,J,K)-HI(I,J+2)*U(I,J+2,K)*V(I,J+2,K))/(2*DY)
24 GO TO 50
25 33 CONTINUE
26 DIHUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
27 C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
28 GO TO 50
29 34 CONTINUE
30 DIHUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
31 C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
32 GO TO 50
33 35 CONTINUE
34 DIHUVY=(3*HI(I,J)*U(I,J,K)*V(I,J,K)-4*HI(I,J-1)*U(I,J-1,K)
35 C*V(I,J-1,K)*HI(I,J-2)+U(I,J-2,K)*V(I,J-2,K))/(2*DY)
36 GO TO 50
37 36 CONTINUE
38 DIHUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
39 C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
40 GO TO 50
41 37 CONTINUE
42 DIHUVY=(4*HI(I,J+1)*U(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*
43 CU(I,J,K)*V(I,J,K)-HI(I,J+2)*U(I,J+2,K)*V(I,J+2,K))/(2*DY)
44 GO TO 50
45 38 CONTINUE
46 DIHUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
47 C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
48 GO TO 50
49 39 CONTINUE
50 DIHUVY=(4*HI(I,J+1)*U(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*
51 CU(I,J,K)*V(I,J,K)-HI(I,J+2)*U(I,J+2,K)*V(I,J+2,K))/(2*DY)
52 GO TO 50
53 40 CONTINUE
54 DIHUVY=(3*HI(I,J)*U(I,J,K)*V(I,J,K)-4*HI(I,J-1)*U(I,J-1,K)
55 C*V(I,J-1,K)*HI(I,J-2)+U(I,J-2,K)*V(I,J-2,K))/(2*DY)
56 50 CONTINUE

```

ORIGINAL PAGE IS  
POOR QUALITY

57  
58RETURN  
END

## 9.2.9 DVVY

This program computes DIHVY. This program is called by INTE.  $\frac{\partial}{\partial \beta}$  (hvv) is computed for interior, boundary or corner by the scheme similar to the one used in the subroutine DINERU.

\*DOC.DVVY

```

1      SUBROUTINE DVVY(I,J,K,IN,JN,KN,U,V,HI,DY,D1HVVY,MAR)
2      DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),MAR(IN,JN)
3      IF(MAR(I,J).EQ.0) GO TO 50
4      IF(MAR(I,J).EQ.1) GO TO 31
5      IF(MAR(I,J).EQ.2) GO TO 32
6      IF(MAR(I,J).EQ.3) GO TO 33
7      IF(MAR(I,J).EQ.4) GO TO 34
8      IF(MAR(I,J).EQ.5) GO TO 35
9      IF(MAR(I,J).EQ.6) GO TO 36
10     IF(MAR(I,J).EQ.7) GO TO 37
11     IF(MAR(I,J).EQ.8) GO TO 38
12     IF(MAR(I,J).EQ.9) GO TO 39
13     IF(MAR(I,J).EQ.10) GO TO 40
14     D1HVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
15     CV(I,J-1,K)*HI(I,J-1))/(2*DY)
16     GO TO 50
17     31 CONTINUE
18     D1HVVY=(3*HI(I,J)*V(I,J,K)*V(I,J,K)+HI(I,J-2)*V(I,J-2,K)
19     C*V(I,J-2,K)-4*HI(I,J-1)*V(I,J-1,K)*V(I,J-1,K))/(2*DY)
20     GO TO 50
21     32 CONTINUE
22     D1HVVY=(4*HI(I,J+1)*V(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*V(I,J,K)
23     C*V(I,J,K)-HI(I,J+2)*V(I,J+2,K)*V(I,J+2,K))/(2*DY)
24     GO TO 50
25     33 CONTINUE
26     D1HVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
27     CV(I,J-1,K)*HI(I,J-1))/(2*DY)
28     GO TO 50
29     34 CONTINUE
30     D1HVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
31     CV(I,J-1,K)*HI(I,J-1))/(2*DY)
32     GO TO 50
33     35 CONTINUE
34     D1HVVY=(3*HI(I,J)*V(I,J,K)*V(I,J,K)+HI(I,J-2)*V(I,J-2,K)
35     C*V(I,J-2,K)-4*HI(I,J-1)*V(I,J-1,K)*V(I,J-1,K))/(2*DY)
36     GO TO 50
37     36 CONTINUE
38     D1HVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
39     CV(I,J-1,K)*HI(I,J-1))/(2*DY)
40     GO TO 50
41     37 CONTINUE
42     D1HVVY=(4*HI(I,J+1)*V(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*V(I,J,K)
43     C*V(I,J,K)-HI(I,J+2)*V(I,J+2,K)*V(I,J+2,K))/(2*DY)
44     GO TO 50
45     38 CONTINUE
46     D1HVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
47     CV(I,J-1,K)*HI(I,J-1))/(2*DY)
48     GO TO 50
49     39 CONTINUE
50     D1HVVY=(4*HI(I,J+1)*V(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*V(I,J,K)
51     C*V(I,J,K)-HI(I,J+2)*V(I,J+2,K)*V(I,J+2,K))/(2*DY)
52     GO TO 50
53     40 CONTINUE
54     D1HVVY=(3*HI(I,J)*V(I,J,K)*V(I,J,K)+HI(I,J-2)*V(I,J-2,K)
55     C*V(I,J-2,K)-4*HI(I,J-1)*V(I,J-1,K)*V(I,J-1,K))/(2*DY)
56     50 CONTINUE

```

57  
58RETURN  
END



## 9.2.10 DIVSU

This subroutine computes DIUX, D2UX, DIUY. This program is called by INTE.  $\frac{\partial u}{\partial \alpha}$  and  $\frac{\partial u}{\partial \beta}$  are computed for interior, boundary or corner points by a scheme similar to the one used in DINERU.

\*DOC.DVISU

```

1  SUBROUTINE DVISU(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1UX,D2UX,D1UY,D2UY,
2  CHAR)
3  DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),MAR(IN,JN)
4  IF(MAR(I,J).EQ.C) GO TO 50
5  IF(MAR(I,J).EQ.1) GO TO 31
6  IF(MAR(I,J).EQ.2) GO TO 32
7  IF(MAR(I,J).EQ.3) GO TO 33
8  IF(MAR(I,J).EQ.4) GO TO 34
9  IF(MAR(I,J).EQ.5) GO TO 35
10 IF(MAR(I,J).EQ.6) GO TO 36
11 IF(MAR(I,J).EQ.7) GO TO 37
12 IF(MAR(I,J).EQ.8) GO TO 38
13 IF(MAR(I,J).EQ.9) GO TO 39
14 IF(MAR(I,J).EQ.10) GO TO 40
15 D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
16 D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
17 D2UX=(U(I+1,J,K)-2*U(I,J,K)+U(I-1,J,K))/(DX*DX)
18 D2UY=(U(I,J+1,K)-2*U(I,J,K)+U(I,J-1,K))/(DY*DY)
19 GO TO 50
20 31 CONTINUE
21 D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
22 D2UX=(U(I+1,J,K)-2*U(I,J,K)+U(I-1,J,K))/(DX*DX)
23 D1UY=(3*U(I,J,K)+U(I,J-2,K)-4*U(I,J-1,K))/(2*DY)
24 D2UY=(U(I,J,K)+U(I,J-2,K)-2*U(I,J-1,K))/(DY*DY)
25 GO TO 50
26 32 CONTINUE
27 D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
28 D2UX=(U(I+1,J,K)-2*U(I,J,K)+U(I-1,J,K))/(DX*DX)
29 D1UY=(4*U(I,J+1,K)-3*U(I,J,K)-U(I,J+2,K))/(2*DY)
30 D2UY=(U(I,J+2,K)+U(I,J,K)-2*U(I,J+1,K))/(DY*DY)
31 GO TO 50
32 33 CONTINUE
33 D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
34 D2UY=(U(I,J+1,K)-2*U(I,J,K)+U(I,J-1,K))/(DY*DY)
35 D1UX=(4*U(I+1,J,K)-3*U(I,J,K)-U(I+2,J,K))/(2*DX)
36 D2UX=(U(I+2,J,K)-2*U(I+1,J,K)+U(I,J,K))/(DX*DX)
37 GO TO 50
38 34 CONTINUE
39 D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
40 D2UY=(U(I,J+1,K)-2*U(I,J,K)+U(I,J-1,K))/(DY*DY)
41 D1UX=(3*U(I,J,K)-4*U(I-1,J,K)+U(I-2,J,K))/(2*DX)
42 D2UX=(U(I,J,K)-2*U(I-1,J,K)+U(I-2,J,K))/(DX*DX)
43 GO TO 50
44 35 CONTINUE
45 D1UY=(3*U(I,J,K)+U(I,J-2,K)-4*U(I,J-1,K))/(2*DY)
46 D2UY=(U(I,J,K)+U(I,J-2,K)-2*U(I,J-1,K))/(DY*DY)
47 D1UX=(4*U(I+1,J,K)-3*U(I,J,K)-U(I+2,J,K))/(2*DX)
48 D2UX=(U(I+2,J,K)-2*U(I+1,J,K)+U(I,J,K))/(DX*DX)
49 GO TO 50
50 36 CONTINUE
51 D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
52 D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
53 D2UX=(U(I+1,J,K)-2*U(I,J,K)+U(I-1,J,K))/(DX*DX)
54 D2UY=(U(I,J+1,K)-2*U(I,J,K)+U(I,J-1,K))/(DY*DY)
55 GO TO 50
56 37 CONTINUE

```

```

57      D1UY=(4*U(I,J+1,K)-3*U(I,J,K)-U(I,J+2,K))/(2*DY)
58      D2UY=(U(I,J+2,K)+U(I,J,K)-2*U(I,J+1,K))/(DY*DY)
59      D1UX=(4*U(I+1,J,K)-3*U(I,J,K)-U(I+2,J,K))/(2*DX)
60      D2UX=(U(I+2,J,K)-2*U(I+1,J,K)+U(I,J,K))/(DX*DX)
61      GO TO 50
62      38 CONTINUE
63      D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
64      D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
65      D2UX=(U(I+1,J,K)-2*U(I,J,K)+U(I-1,J,K))/(DX*DX)
66      D2UY=(U(I,J+1,K)-2*U(I,J,K)+U(I,J-1,K))/(DY*DY)
67      GO TO 50
68      39 CONTINUE
69      D1UY=(4*U(I,J+1,K)-3*U(I,J,K)-U(I,J+2,K))/(2*DY)
70      D2UY=(U(I,J+2,K)+U(I,J,K)-2*U(I,J+1,K))/(DY*DY)
71      D1UX=(3*U(I,J,K)-4*U(I-1,J,K)+U(I-2,J,K))/(2*DX)
72      D2UX=(U(I,J,K)-2*U(I-1,J,K)+U(I-2,J,K))/(DX*DX)
73      GO TO 50
74      40 CONTINUE
75      D1UY=(3*U(I,J,K)+U(I,J-2,K)-4*U(I,J-1,K))/(2*DY)
76      D2UY=(U(I,J,K)+U(I,J-2,K)-2*U(I,J-1,K))/(DY*DY)
77      D1UX=(3*U(I,J,K)-4*U(I-1,J,K)+U(I-2,J,K))/(2*DX)
78      D2UX=(U(I,J,K)-2*U(I-1,J,K)+U(I-2,J,K))/(DX*DX)
79      50 CONTINUE
80      RETURN
81      END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

## 9.2.11 DWISV

This program computes DIVY, D2VY, DIVX AND D2VX. This program is called by subroutine INTE. Schemes used are similar to the one used in DVISU.

EN\*DOC.DVISV

```

1      SUBROUTINE DVISV(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1VX,D2VX,D1VY,D2VY,
2      CHAR)
3      DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),MAR(IN,JN)
4      IF(MAR(I,J).EQ.C) GO TO 50
5      IF(MAR(I,J).EQ.1) GO TO 31
6      IF(MAR(I,J).EQ.2) GO TO 32
7      IF(MAR(I,J).EQ.3) GO TO 33
8      IF(MAR(I,J).EQ.4) GO TO 34
9      IF(MAR(I,J).EQ.5) GO TO 35
10     IF(MAR(I,J).EQ.6) GO TO 36
11     IF(MAR(I,J).EQ.7) GO TO 37
12     IF(MAR(I,J).EQ.8) GO TO 38
13     IF(MAR(I,J).EQ.9) GO TO 39
14     IF(MAR(I,J).EQ.10) GO TO 40
15     D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
16     D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
17     D2VX=(V(I+1,J,K)-2*V(I,J,K)+V(I-1,J,K))/(DX*DX)
18     D2VY=(V(I,J+1,K)-2*V(I,J,K)+V(I,J-1,K))/(DY*DY)
19     GO TO 50
20     31 CONTINUE
21     D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
22     D2VX=(V(I+1,J,K)-2*V(I,J,K)+V(I-1,J,K))/(DX*DX)
23     D1VY=(3*V(I,J,K)-4*V(I,J-1,K)+V(I,J-2,K))/(2*DY)
24     D2VY=(V(I,J,K)+V(I,J-2,K)-2*V(I,J-1,K))/(DY*DY)
25     GO TO 50
26     32 CONTINUE
27     D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
28     D2VX=(V(I+1,J,K)-2*V(I,J,K)+V(I-1,J,K))/(DX*DX)
29     D1VY=(4*V(I,J+1,K)-3*V(I,J,K)-V(I,J+2,K))/(2*DY)
30     D2VY=(V(I,J+2,K)+V(I,J,K)-2*V(I,J+1,K))/(DY*DY)
31     GO TO 50
32     33 CONTINUE
33     D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
34     D2VY=(V(I,J+1,K)-2*V(I,J,K)+V(I,J-1,K))/(DY*DY)
35     D1VX=(4*V(I+1,J,K)-3*V(I,J,K)-V(I+2,J,K))/(2*DX)
36     D2VX=(V(I+2,J,K)-2*V(I+1,J,K)+V(I,J,K))/(DX*DX)
37     GO TO 50
38     34 CONTINUE
39     D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
40     D2VY=(V(I,J+1,K)-2*V(I,J,K)+V(I,J-1,K))/(DY*DY)
41     D1VX=(3*V(I,J,K)-4*V(I-1,J,K)+V(I-2,J,K))/(2*DX)
42     D2VX=(V(I,J,K)-2*V(I-1,J,K)+V(I-2,J,K))/(DX*DX)
43     GO TO 50
44     35 CONTINUE
45     D1VY=(3*V(I,J,K)-4*V(I,J-1,K)+V(I,J-2,K))/(2*DY)
46     D2VY=(V(I,J,K)+V(I,J-2,K)-2*V(I,J-1,K))/(DY*DY)
47     D1VX=(4*V(I+1,J,K)-3*V(I,J,K)-V(I+2,J,K))/(2*DX)
48     D2VX=(V(I+2,J,K)-2*V(I+1,J,K)+V(I,J,K))/(DX*DX)
49     GO TO 50
50     36 CONTINUE
51     D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
52     D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
53     D2VX=(V(I+1,J,K)-2*V(I,J,K)+V(I-1,J,K))/(DX*DX)
54     D2VY=(V(I,J+1,K)-2*V(I,J,K)+V(I,J-1,K))/(DY*DY)
55     GO TO 50
56     37 CONTINUE

```

```

57      DIVY=(4*V(I,J+1,K)-3*V(I,J,K)-V(I,J+2,K))/(2*DY)
58      D2VY=(V(I,J+2,K)+V(I,J,K)-2*V(I,J+1,K))/(DY*DY)
59      D1VX=(4*V(I+1,J,K)-3*V(I,J,K)-V(I+2,J,K))/(2*DX)
60      D2VX=(V(I+2,J,K)-2*V(I+1,J,K)+V(I,J,K))/(DX*DX)
61      GO TO 50
62 38      CONTINUE
63      D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
64      D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
65      D2VX=(V(I+1,J,K)-2*V(I,J,K)+V(I-1,J,K))/(DX*DX)
66      D2VY=(V(I,J+1,K)-2*V(I,J,K)+V(I,J-1,K))/(DY*DY)
67      GO TO 50
68 39      CONTINUE
69      DIVY=(4*V(I,J+1,K)-3*V(I,J,K)-V(I,J+2,K))/(2*DY)
70      D2VY=(V(I,J+2,K)+V(I,J,K)-2*V(I,J+1,K))/(DY*DY)
71      D1VX=(3*V(I,J,K)-4*V(I-1,J,K)+V(I-2,J,K))/(2*DX)
72      D2VX=(V(I,J,K)-2*V(I-1,J,K)+V(I-2,J,K))/(DX*DX)
73      GO TO 50
74 40      CONTINUE
75      D1VY=(3*V(I,J,K)-4*V(I,J-1,K)+V(I,J-2,K))/(2*DY)
76      D2VY=(V(I,J,K)+V(I,J-2,K)-2*V(I,J-1,K))/(DY*DY)
77      D1VX=(3*V(I,J,K)-4*V(I-1,J,K)+V(I-2,J,K))/(2*DX)
78      D2VX=(V(I,J,K)-2*V(I-1,J,K)+V(I-2,J,K))/(DX*DX)
79 50      CONTINUE
80      RETURN
81      END

```

**9.2.12 D12Z**

This subroutine computes D1UZ, DZUZ, D1VZ, DZVZ, D1A3Z, DIUVZ. This subroutine is called by INTE. At points on the surface, KN=1, forward difference scheme is employed.

\*DOC.D12Z

```

1 C *****
2 C THIS PROGRAM CALCULATES THE Z DERIVATIVES
3 C *****
4 SUBROUTINE D12Z (I,J,K,IN,JN,KN,U,V,W,HI,HX,HV,DX,DY,DZ,D1UWZ,
5 CA3,TAUX,TAUY,
6 CD1VMZ,D1UZ,D2UZ,D1VZ,D2VZ,D1A3Z)
7 DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),HI(IN,JN),
8 CHX(IN,JN),HY(IN,JN)
9 DIMENSION A3(KN)
10 IF (K.EQ.1) GO TO 61
11 IF (K.EQ.KN) GO TO 62
12 D1UZ=(U(I,J,K+1)-U(I,J,K-1))/(2*DZ)
13 D1VZ=(V(I,J,K+1)-V(I,J,K-1))/(2*DZ)
14 D2UZ=(U(I,J,K+1)-2*U(I,J,K)+U(I,J,K-1))/(DZ*DZ)
15 D2VZ=(V(I,J,K+1)-2*V(I,J,K)+V(I,J,K-1))/(DZ*DZ)
16 D1A3Z=(A3(K+1)-A3(K-1))/(2*DZ)
17 D1UWZ=(U(I,J,K+1)*W(I,J,K+1)-U(I,J,K-1)*W(I,J,K-1))/(2*DZ)
18 D1VMZ=(V(I,J,K+1)*W(I,J,K+1)-V(I,J,K-1)*W(I,J,K-1))/(2*DZ)
19 GO TO 63
20 61 CONTINUE
21 D1UZ=HI(I,J)*TAUX
22 D1VZ=HI(I,J)*TAUY
23 D2UZ=2*(U(I,J,K+1)-U(I,J,K))/(DZ*DZ)-2*(TAUX*HI(I,J)/DZ)
24 D2VZ=2*(V(I,J,K+1)-V(I,J,K))/(DZ*DZ)-2*(TAUY*HI(I,J)/DZ)
25 D1A3Z=(4*A3(K+1)-3*A3(K)-A3(K-2))/(2*DZ)
26 D1UWZ=(4*U(I,J,K+1)*W(I,J,K+1)-3*U(I,J,K)*W(I,J,K)-U(I,J,K-2)*W(I
27 C,J,K+2))/(2*DZ)
28 D1VMZ=(4*V(I,J,K+1)*W(I,J,K+1)-3*V(I,J,K)*W(I,J,K)
29 C-V(I,J,K-2)*W(I,J,K-2))/(2*DZ)
30 GO TO 63
31 62 CONTINUE
32 D1UZ=(3*U(I,J,K)-4*U(I,J,K-1)+U(I,J,K-2))/(2*DZ)
33 D1VZ=(3*V(I,J,K)-4*V(I,J,K-1)+V(I,J,K-2))/(2*DZ)
34 D2UZ=(U(I,J,K-2)+U(I,J,K)-2*U(I,J,K-1))/(DZ*DZ)
35 D2VZ=(V(I,J,K-2)+V(I,J,K)-2*V(I,J,K-1))/(DZ*DZ)
36 D1A3Z=(3*A3(K)-4*A3(K-1)+A3(K-2))/(2*DZ)
37 D1UWZ=(3*U(I,J,K)*W(I,J,K)-4*U(I,J,K-1)*W(I,J,K-1)
38 C+U(I,J,K-2)*W(I,J,K-2))/(2*DZ)
39 D1VMZ=(3*V(I,J,K)*W(I,J,K)-4*V(I,J,K-1)*W(I,J,K-1)
40 C+V(I,J,K-2)*W(I,J,K-2))/(2*DZ)
41 63 CONTINUE
42 RETURN
43 END

```

ORIGINAL PAGE IS  
OF POOR QUALITY



92.13 ERROR: Calculates "HIRT and HARLOW" correction term at half gridpoints and at the surface. The last term in Poisson's equation is "HIRT and HARLOW" correction term. This is evaluated by a backward difference in time with present time set equal to zero. This is necessary because Poisson's equation is usually solved by the iterative technique usually leading to errors. If they are not corrected, continuity equation will not be satisfied leading to accumulation or loss of fluid from the system.

In the program

$$\text{WHLDT (IW, JW)} = \frac{-\text{WH(IW,JW,1)}}{\text{DT}}$$

WH at previous time step is set at zero.

WH at present time step is nonzero.

\*\*\*\*DOC,ERROR

```

1  C*****
2  C    THIS PROGRAM CALCULATES THE HIRT AND HARLOW CORRECTION TERM AT THE
3  C    SURFACE
4  C*****
5  SUBROUTINE ERROR(IWN,JWN, IW ,JW,DT,WH,WHLDT,KN,MRH)
6  DIMENSION WHLDT(IWN,JWN),WH(IWN,JWN,KN)
7  DIMENSION MRH(IWN,JWN)
8  C WHLDT IS THE TIME DERIVATIVE OF W AT HALF GRID POINTS AT LID
9  DO 3100 IW=1,IWN
10 DO 3100 JW=1,JWN
11 IF (MRH(IW,JW).EQ.C) GO TO 3000
12 WHLDT(IW,JW)=-WH(IW,JW,1)/DT
13 3000 CONTINUE
14 3100 CONTINUE
15 RETURN
16 END

```

9.2.14 FORCE: This program computes R.H.S. of Poisson's equation at half grid points.

$$\begin{aligned}
 FH = & \frac{1}{h} \frac{\partial}{\partial \alpha} (-Ax_1 + Ax_2 + C_x - Xp) \\
 & + \frac{1}{h} \frac{\partial}{\partial \beta} (-Ay_1 - Ay_2 + C_y - Yp) \\
 & - \frac{1}{h} \left( \frac{\partial h}{\partial \alpha} \frac{\partial Ps}{\partial \alpha} + \frac{\partial h}{\partial \beta} \frac{\partial Ps}{\partial \beta} \right) - \frac{\partial \Omega}{\partial t} \Big|_z = 0 \\
 = & \frac{1}{h} \frac{\partial}{\partial \alpha} (XINT) + \frac{1}{h} \frac{\partial}{\partial \beta} (YINT) \\
 & - \frac{1}{h} \left[ \frac{\partial h}{\partial \alpha} \left( DPSX + \frac{1}{Rb} v - Bx \right) + \frac{\partial h}{\partial \beta} \left( DPSY - \frac{1}{Rb} u - By \right) \right] \\
 & - WHLDT
 \end{aligned}$$

This program takes  $\frac{1}{Rb} v$ ,  $\frac{1}{Rb} u$ ,  $Bx$  and  $By$  equal to zero

$$\frac{\partial Ps}{\partial \alpha} = DPSX$$

and  $\frac{\partial Ps}{\partial \alpha}$  at half grid points is average of four surrounding main grid points.

$\frac{\partial}{\partial \alpha} (XINT)$ ,  $\frac{\partial}{\partial \beta} (YINT)$  at half grid points is average of four surrounding main grid points.

10000.FORCE

```

1  SUBROUTINE FORCE(I,J,IN,JN,XINT,YINT,WHLDT,DX,DY,HI,HX,HY,
2  CHRH,
3  CDPSX,DPSY,FH,AP,IN,JN,IWN,JWN,RINTX,RINTY,U,V,EUL,ABR,MAR,MN)
4  DIMENSION XINT(IN,JN),YINT(IN,JN),WHLDT(IWN,JWN),HI(IN,JN),HX(IN,
5  C,JN),HY(IN,JN),DPSX(IN,JN),DPSY(IN,JN),FH(IWN,JWN)
6  DIMENSION MRH(IWN,JWN)
7  DIMENSION RINTX(IN,JN,KN),RINTY(IN,JN,KN),U(IN,JN,KN),V(IN,JN,KN)
8  C,MAR(IN,JN)
9  K=1
10 DO 10 I=1,IN
11 DO 10 J=1,JN
12 IF(MAR(I,J).LT.11) GO TO 90
13 DPSX(I,J)=DPSX(I,J)-EUL*RINTX(I,J,K)+V(I,J,K)*ABR
14 DPSY(I,J)=DPSY(I,J)-EUL*RINTY(I,J,K)-U(I,J,K)*ABR
15 90 CONTINUE
16 DO 10 IW=1,IWN
17 DO 10 JW=1,JWN
18 I=IW
19 J=JW
20 IF(MRH(IW,JW).EQ.0) GO TO 9
21 DPSXH=(DPSX(I,J)+DPSX(I+1,J)+DPSX(I,J+1)+DPSX(I+1,J+1))/4.0
22 DPSYH=(DPSY(I,J)+DPSY(I+1,J)+DPSY(I,J+1)+DPSY(I+1,J+1))/4.0
23 HXH=(HX(I,J)+HX(I+1,J)+HX(I,J+1)+HX(I+1,J+1))/4.0
24 HYH=(HY(I,J)+HY(I+1,J)+HY(I,J+1)+HY(I+1,J+1))/4.0
25 DXINT=(XINT(I+1,J)+XINT(I+1,J+1)-XINT(I,J)-XINT(I,J+1))/(2*DX)
26 DYINT=(YINT(I,J+1)+YINT(I+1,J+1)-YINT(I,J)-YINT(I+1,J))/(2*DY)
27 HH=(HI(I,J)+HI(I+1,J)+HI(I,J+1)+HI(I+1,J+1))/4.0
28 FH(IW,JW)=(1./AP)*(-(1./HH)*(DXINT+DYINT)-WHLDT(IW,JW)-(AP/HH)*
29 C(HXH*DPSXH+HYH*DPSYH))
30 9 CONTINUE
31 10 CONTINUE
32 RETURN
33 END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

## 9.2.15 GRADS

This program compute slopes of the bottom using non-dimensionalized and unstretched depths and  $\alpha$  and  $\beta$  coordinates.

M\*DULL(1).GRADS

```

1  SUBROUTINE GRADS(IN,JN,KN,IWN,JWN,HI,HX,H7,MAR,MRH,DX,DY)
2  DIMENSION MAR(IN,JN),HI(IN,JN),HX(IN,JN),HY(IN,JN)
3  DIMENSION MRH(IWN,JWN)
4  C      TO CALCULATE HX AND HY
5  DO 50 I=1,IN
6  DO 50 J=1,JN
7  IF(MAR(I,J).EQ.0) GO TO 50
8  IF(MAR(I,J).EQ.1) GO TO 31
9  IF(MAR(I,J).EQ.2) GO TO 32
10 IF(MAR(I,J).EQ.3) GO TO 33
11 IF(MAR(I,J).EQ.4) GO TO 34
12 IF(MAR(I,J).EQ.5) GO TO 35
13 IF(MAR(I,J).EQ.6) GO TO 36
14 IF(MAR(I,J).EQ.7) GO TO 37
15 IF(MAR(I,J).EQ.8) GO TO 38
16 IF(MAR(I,J).EQ.9) GO TO 39
17 IF(MAR(I,J).EQ.10) GO TO 40
18 HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2*DX)
19 HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2*DY)
20 GO TO 50
21 31 CONTINUE
22 HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2*DX)
23 HY(I,J)=(3*HI(I,J)+HI(I,J-2)-4*HI(I,J-1))/(2*DY)
24 GO TO 50
25 32 CONTINUE
26 HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2*DX)
27 HY(I,J)=(4*HI(I,J+1)-3*HI(I,J)-HI(I,J+2))/(2*DY)
28 GO TO 50
29 33 CONTINUE
30 HX(I,J)=(4*HI(I+1,J)-3*HI(I,J)-HI(I+2,J))/(2*DX)
31 HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2*DY)
32 GO TO 50
33 34 CONTINUE
34 HX(I,J)=(2*HI(I,J)+HI(I-2,J)-4*HI(I-1,J))/(2*DX)
35 HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2*DY)
36 GO TO 50
37 35 CONTINUE
38 HX(I,J)=(4*HI(I+1,J)-3*HI(I,J)-HI(I+2,J))/(2*DX)
39 HY(I,J)=(3*HI(I,J)+HI(I,J-2)-4*HI(I,J-1))/(2*DY)
40 GO TO 50
41 36 CONTINUE
42 HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2*DX)
43 HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2*DY)
44 GO TO 50
45 37 CONTINUE
46 HX(I,J)=(4*HI(I+1,J)-3*HI(I,J)-HI(I+2,J))/(2*DX)
47 HY(I,J)=(4*HI(I,J+1)-3*HI(I,J)-HI(I,J+2))/(2*DY)
48 GO TO 50
49 38 CONTINUE
50 HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2*DX)
51 HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2*DY)
52 GO TO 50
53 39 CONTINUE
54 HX(I,J)=(3*HI(I,J)+HI(I-2,J)-4*HI(I-1,J))/(2*DX)
55 HY(I,J)=(4*HI(I,J+1)-3*HI(I,J)-HI(I,J+2))/(2*DY)
56 GO TO 50

```

```

57      40      CONTINUE
58      HX(I,J)=(3*HI(I,J)+HI(I-2,J)-4*HI(I-1,J))/(2*DX)
59      HY(I,J)=(3*HI(I,J)+HI(I,J-2)-4*HI(I,J-1))/(2*DY)
60      50      CONTINUE
61      WRITE (9) ((MAR(I,J),I=1,IN),J=1,JN),
62      C((MRH(IW,JW),IW=1,IWN),JW=1,JWN),
63      C((HI(I,J),I=1,IN),J=1,JN),
64      DO 60 I=1,IN
65      PRINT 61,I,(MAR(I,J),J=1,JN)
66      61      FORMAT(/' I=',I3/, ' MARKER'/(5X,9I3))
67      60      CONTINUE
68      DO 62 IW=1,IWN
69      PRINT 63,IW,(MRH(IW,JW),JW=1,JWN)
70      63      FORMAT(/' IW=',I3/, ' MIDMARKER'/(5X,8I3))
71      62      CONTINUE
72      DO 70 I=1,IN
73      PRINT 71,I,(HI(I,J),J=1,JN)
74      71      FORMAT(/' I=',I3/, ' DEPTH'/(5X,9E14.7))
75      PRINT 72,I,(HX(I,J),J=1,JN)
76      72      FORMAT(/' I=',I3/, ' XGRAD'/(5X,9E14.7))
77      PRINT 73,I,(HY(I,J),J=1,JN)
78      73      FORMAT(/' I=',I3/, ' YGRAD'/(5X,9E14.7))
79      70      CONTINUE
80      END

```

PRECEDING PAGE BLANK NOT FILMED

ORIGINAL PAGE IS  
OF POOR QUALITY

## 9.2.16 HEIGHT

This program inputs depth of the basin into the model.

This subroutine is for constant depth model.

HI (I,J) = CC = A constant and non-dimensionalized depth = 1.0

HX (I,J) = 0.0 x derivative

HY (I,J) = 0.0 y derivative



N=DOC,HEIGHT

1 C  
 2  
 3  
 4  
 5  
 6  
 7  
 8  
 9 100  
 10  
 11

THIS PROGRAM PUTS CONSTANT DEPTH FOR CC=1.0 IN THE DATA  
 SUBROUTINE HEIGHT(I,J,K,IN,JN,KN,HI,HX,HY,CC)  
 DIMENSION HI(IN,JN),HX(IN,JN),HY(IN,JN)  
 DO 100 I=1,IN  
 DO 100 J=1,JN  
 HI(I,J)=CC  
 HX(I,J)=0.0  
 HY(I,J)=0.0  
 CONTINUE  
 RETURN  
 END

9.2.17 HEIGH1

This subroutine reads the depths for the near field model in the format given.

\*DOC.HEIGH1

```

1      C      THIS PROGRAM READS DEPTHS FROM THE DATA
2      SUBROUTINE HEIGH1(I,J,K,IN,JN,KN,HI,HX,HY,CC,JX)
3      DIMENSION HI(IN,JN),HX(IN,JN),HY(IN,JN),JX(JN)
4      DO 100 I=1,IN
5      DO 100 J=1,JN
6      HI(I,J)=CC
7      HX(I,J)=0.0
8      HY(I,J)=0.0
9      100    CONTINUE
10     DO 200 JJ=1,JN
11     READ 2,J,(HI(I,J),I=1,IN)
12     PRINT 3,J,(HI(I,J),I=1,IN)
13     JX(JJ)=J
14     200    CONTINUE
15     3      FORMAT(5X,I5,18F6.2)
16     2      FORMAT()
17     RETURN
18     END

```

#### 9.2.18 HITEA

This subroutine is used by the far field stratified and unstratified models. The subroutine sets the depth everywhere equal to 1 if cc is nonzero, otherwise the depth matrix HI is read in from the data element DATAML when the main program TMAIN4 or TMAIN4CB is executed.

4\*DJLL(1).HITEA

```

1      SUBROUTINE HITEA(I,J,K,IN,JN,KN,HI,HX,HY,CC)
2      DIMENSION HI(IN,JN),HX(IN,JN),HY(IN,JN)
3      IF(CC.LT.C.0001) GO TO 150
4      DO 100 I=1,IN
5      DO 100 J=1,JN
6          HI(I,J)=CC
7          HX(I,J)=0.3
8          HY(I,J)=0.3
9      100 CONTINUE
10     GO TO 250
11     150 CONTINUE
12     DO 200 II=1,IN
13     READ 2,I,(HI(I,J),J=1,JN)
14     PRINT 3,I,(HI(I,J),J=1,JN)
15     200 CONTINUE
16     250 CONTINUE
17     3  FORMAT(5X,I5,13F6.2)
18     2  FORMAT( )
19     RETURN
20     END

```

## 9.2.19 INITB

This procedure is used for initializing the temperature field for a far field stratified model. The subroutine sets the temperature everywhere equal to the ambient temperature profile defined by the matrix AMINT (NTL, NTLV). The first column in this matrix is the depths and the second column are the corresponding temperatures.

M\*DULL(1).INITB

```

1 C*****
2 C      THIS PROGRAM INITIALIZES TEMP AND DENSITY
3 C*****
4 SUBROUTINE INITB(I,J,K,IN,JN,KN,IW,JW,INN,JWN,A,B,C,T,RO,
5   CHAR,MRH,
6   CTREF,RREF,
7   CTO,AMINT,HI,NTL,NTLV)
8   DIMENSION T(IN,JN,KN),RO(IN,JN,KN)
9   DIMENSION MAR(IN,JN),MRH(IW,JW),AMINT(NTL,NTLV),HI(IN,JN)
10  TOU=(TO-TREF)/TREF
11  R=A+B*TO+C*TO*TO
12  ROD=(R-RREF)/RREF
13  DO 10 I=1,IN
14  DO 10 J=1,JN
15  IF (MAR(I,J).EQ.D) GO TO 12
16  DO 11 K=1,KN
17  HIK=(K-1)*HI(I,J)/(KN-1)
18  NTL=NTL-1
19  DO 100 N=1,NTLM
20  IF(HIK.GE.AMINT(N,1).AND.HIK.LT.AMINT(N+1,1))
21  CT(I,J,K)=AMINT(N,2)
22  100 CONTINUE
23  IF(HIK.GE.AMINT(NTL,1))
24  CT(I,J,K)=AMINT(NTL,2)
25  RO(I,J,K)=A+B*T(I,J,K)+C*T(I,J,K)**2
26  T(I,J,K)=(T(I,J,K)-TREF)/TREF
27  RO(I,J,K)=(RO(I,J,K)-RREF)/RREF
28  11 CONTINUE
29  12 CONTINUE
30  10 CONTINUE
31  RETURN
32  END

```

## 9.2.20 INITIA

This program initializes the values of u,v, w. WH, D, E and PINTH. This program sets u, v, w, D, E. and wH equal to zero. PINTH is set equal to ARBP.



## DOC.INITIA

```

1  C*****
2  C   THIS PROGRAM INITIALIZES THE VALUES OF U,V,W,H,M,D,E,PINTH
3  C*****
4  SUBROUTINE INITIA(IN,JN,KN,IWN,JWN,U,V,W,H,M,D,E,PINTH,I,J,K,IW,JW,
5  CARBP),
6  DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),H(IWN,JWN,KN),
7  CD(IN,JN,KN),E(IN,JN,KN),
8  CPINTH(IWN,JWN)
9  C INITIAL CONDITIONS ON U AND V
10 DO 100 I=1,IN
11 DO 100 J=1,JN
12 DO 100 K=1,KN
13   U(I,J,K)=C
14   V(I,J,K)=0
15   W(I,J,K)=0
16   D(I,J,K)=C.0
17   E(I,J,K)=C.0
18 100 CONTINUE
19 C INITIAL CONDITIONS ON WH AND PH
20 DO 200 IW=1,IWN
21 DO 200 JW=1,JWN
22   PINTH(IW,JW)=ARBp
23 DO 200 K=1,KN
24   WH(IW,JW,K)=0
25 200 CONTINUE
26 RETURN
27 END

```

## 9.2.21 INITIT

This program sets initial temperature field. It sets the temperature field to the reference temperature at all points.

## DOC,INITIT

```

1  C*****
2  C    THIS PROGRAM INITIALIZES TEMP AND DENSITY
3  C*****
4  SUBROUTINE INITIT(I,J,K,IN,JN,KN,IW,JW,IWN,JWN,A,B,C,T,RO,
5  CHAR,HRH,
6  CTREF,RRREF,
7  CTW,ROW,TO)
8  DIMENSION T(IN,JN,KN),RO(IN,JN,KN),TW(IWN,JWN,KN),ROW(IWN,JWN,KN)
9  DIMENSION HAR(IN,JN),HRH(IWN,JWN)
10 T0=(T0-TREF)/TREF
11 R=A+B*T0+C*T0*T0
12 ROD=(R-RRREF)/RRREF
13 DO 10 I=1,IN
14 DO 10 J=1,JN
15 IF (HAR(I,J).EQ.0) GO TO 12
16 DO 11 K=1,KN
17 T(I,J,K)=T0D
18 RO(I,J,K)=ROD
19 11 CONTINUE
20 12 CONTINUE
21 10 CONTINUE
22 DO 20 IW=1,IWN
23 DO 20 JW=1,JWN
24 IF (HRH(IW,JW).EQ.0) GO TO 22
25 DO 21 K=1,KN
26 TW(IW,JW,K)=TCD
27 ROW(IW,JW,K)=ROD
28 21 CONTINUE
29 22 CONTINUE
30 20 CONTINUE
31 RETURN
32 END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

## 9.2.22 INITM

This subroutine is used by the far field unstratified model. This subroutine is used by the main program TMAIN4T which sets up the temperature field equal to a measured initial value. This subroutine reads in the surface temperature matrix stored as data element ITPK1. Temperature below the surface are computed by assuming a temperature drop of  $1^{\circ}\text{C}$  over the reference depth, a condition which may be changed if desired by making changes in line #21 of this subroutine.

\*DULL(1).INITM

```

1  C*****
2  C      THIS PROGRAM INITIALIZES TEMP AND DENSITY FOR MORNING
3  C*****
4  SUBROUTINE INITM(I,J,K,IN,JN,KN,IN,JN,IWN,JWN,A,B,C,T,RO,
5  CHAR,MRH,
6  CTREF,RREF,
7  CTW,ROW,TC,HI)
8  DIMENSION T(IN,JN,KN),RO(IN,JN,KN),TW(INN,JWN,KN),ROW(IWN,JWN,KN)
9  DIMENSION MRH(IN,JN),MRH(IWN,JWN),HI(IN,JN)
10 TOD=(TO-TREF)/TREF
11 R=A+B*TO+C*TO*TO
12 ROD=(R-RREF)/RREF
13 DO 900 II=1,IN
14 READ 2,1,(T(I,J,1),J=1,JN)
15 900 CONTINUE
16 2 FORMAT (1)
17 DO 1001 K=2,KN
18 DO 1001 J=1,JN
19 DO 1001 I=1,IN
20 DP=HI(I,J)*FLOAT(K-1)/FLOAT(KN-1)
21 T(I,J,K)=T(I,J,1)-DP*1.0
22 1001 CONTINUE
23 DO 1002 K=1,KN
24 DO 1002 J=1,JN
25 DO 1002 I=1,IN
26 RO(I,J,K)=A+B*T(I,J,K)+C*T(I,J,K)**2
27 T(I,J,K)=(T(I,J,K)-TREF)/TREF
28 RO(I,J,K)=(RO(I,J,K)-RREF)/RREF
29 1002 CONTINUE
30 11 CONTINUE
31 12 CONTINUE
32 10 CONTINUE
33 DO 20 IW=1,IWN
34 DO 20 JW=1,JWN
35 IF (MRH(IW,JW).EQ.C) GO TO 22
36 DO 21 K=1,KN
37 TW(IW,JW,K)=T(IW,JW,K)
38 ROW(IW,JW,K)=RO(IW,JW,K)
39 21 CONTINUE
40 22 CONTINUE
41 20 CONTINUE
42 RETURN
43 END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

## 9.2.23 INITMB

This subroutine updates the temperature field for a far field stratified model from the ambient field to the measured temperature field including the thermal plume. The subroutine reads in TSMN (the minimum surface temperature), DPMX (maximum depth to which the effect of thermal plume is extended) and the surface temperature matrix. Temperatures below the surface are computed by assuming a linear accumulation of plume heat from a maximum value at the surface to a zero at DPMX,

Thus:

Temperature at depth  $d$  ( $<DPMX$ ) = Surface Temperature

$$\text{Temp} = \frac{d}{DPMX} (\text{Surface temperature} - \text{TSMN})$$

Temperature at depth  $d$  ( $>DPMX$ ) = Ambient temperature

\*DULL(1).INITMB

```

1 C*****
2 C      THIS PROGRAM INITIALIZES TEMP AND DENSITY FOR MORNING
3 C*****
4 SUBROUTINE INITMB(I,J,K,IN,JN,KN,IN,JN,INN,JN,A,B,C,T,RO,
5 CHAK,MRH,
6 CTREF,RREF,
7 CTO,HI)
8 DIMENSION T(IN,JN,KN),RO(IN,JN,KN)
9 DIMENSION MAR(IN,JN),MRH(ILN,JN),HI(IN,JN)
10 T0=(T0-TREF)/TREF
11 R=A+B*T0+C*T0*T0
12 R0=(R-RREF)/RREF
13 READ 2,TSMN,DPMX
14 DO 900 II=1,IN
15 READ 2,I,(T(I,J,1),J=1,JN)
16 900 CONTINUE
17 2 FORMAT (I)
18 DO 1001 K=2,KN
19 DO 1001 J=1,JN
20 DO 1001 I=1,IN
21 T(I,J,K)=(1.0+T(I,J,K))*TREF
22 DP=HI(I,J)*FLOAT(K-1)/FLOAT(KN-1)
23 IF(DP.LT.DPMX) T(I,J,K)=T(I,J,K)+((DPMX-DP)/DPMX)*(T(I,J,1)-TSMN)
24 1001 CONTINUE
25 DO 1002 K=1,KN
26 DO 1002 J=1,JN
27 DO 1002 I=1,IN
28 RO(I,J,K)=A+B*T(I,J,K)+C*T(I,J,K)**2
29 T(I,J,K)=(T(I,J,K)-TREF)/TREF
30 RO(I,J,K)=(RO(I,J,K)-RREF)/RREF
31 1002 CONTINUE
32 11 CONTINUE
33 12 CONTINUE
34 10 CONTINUE
35 RETURN
36 END

```

#### 9.2.24 INLET

This program inputs the velocities  $u$  and  $v$  at plume discharge into the model. It defines the inlet  $v$ -velocity by two constants  $AA$  and  $BB$ .

$AA$  and  $BB$  are non-dimensionalized numbers. Non dimensionalized with respect to discharge velocity.

Therefore  $AA = 1.0$

$BB = 1.0$



\*DOC-INLET

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13

100

```

SUBROUTINE INLET(I,J,K,IN,JN,KN,V,G,AA,BB)
DIMENSION G(IN,JN,KN),V(IN,JN,KN)
INM1=IN-1
JNM1=JN-1
KNM1=KN-1
DO 100 K=1,KNM1
V(9,I,K)=AA
G(9,I,K)=AA
V(10,I,K)=BB
G(10,I,K)=BB
CONTINUE
RETURN
END

```

### 9.2.25 INLETA

This subroutine reads in the number of inlet and outlet points, u, v and T at inlet points and u and v at outlet points for the far field unstratified model. This subroutine is called in by the main programs TMAIN5, TMAIN5T and TMAIN5V. The subroutine reads in data from element INDATA5, the lines following the first twelve lines.

M=DULL(1).INLETA

```

1      SUBROUTINE INLETA(I,J,K,IN,JN,KN,U,V,H,G,T)
2      DIMENSION H(IN,JN,KN),G(IN,JN,KN),U(IN,JN,KN)
3      DIMENSION V(IN,JN,KN),T(IN,JN,KN)
4      READ 2,NIN,NOUT
5          2      FORMAT(I)
6          20      DO 20 NH=1,NIN
7              READ 2,I,J,K,U(I,J,K),V(I,J,K),T(I,J,K)
8              H(I,J,K)=U(I,J,K)
9              G(I,J,K)=V(I,J,K)
10         20      CONTINUE
11         30      DO 30 NH=1,NOUT
12             READ 2,I,J,K,U(I,J,K),V(I,J,K)
13             H(I,J,K)=U(I,J,K)
14             G(I,J,K)=V(I,J,K)
15         30      CONTINUE
16      RETURN
17      END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

### 9.2.26 INLETB

This subroutine is used for reading in the information at open boundaries for the far field stratified model. The subroutine is the same as INLETA. The subroutine reads in the data from element DATAML5, the lines following the first 13 lines. The subroutine is called in by the main program TMAIN5B, TMAIN5TB and TMAIN5VB.

M=DULL(1).INLETS

```

1      SUBROUTINE INLETS(I,J,K,IN,JN,KN,U,V,H,G,T)
2      DIMENSION H(IN,JN,KN),G(IN,JN,KN),U(IN,JN,KN)
3      DIMENSION V(IN,JN,KN),T(IN,JN,KN)
4      READ 2,NIN,NOUT
5      .4.  FORMAT(1)
6      DO 20 NH=1,NIN
7      READ 2,I,J,K,U(I,J,K),V(I,J,K),T(I,J,K)
8      H(I,J,K)=U(I,J,K)
9      G(I,J,K)=V(I,J,K)
10     .20  CONTINUE
11     DO 30 NM=1,NOUT
12     READ 2,I,J,K,U(I,J,K),V(I,J,K)
13     H(I,J,K)=U(I,J,K)
14     G(I,J,K)=V(I,J,K)
15     .30  CONTINUE
16     RETURN
17     END

```

## 9.2.27 INTE

This subroutine computes XINT, YINT, DPSX and DPSY.

This subroutine uses x-momentum equation to compute  $\frac{\partial Ps}{\partial x}$ .

$$\frac{\partial Ps}{\partial \alpha} = DPSX + \frac{1}{R_b} v - B_x$$

$$\frac{\partial Ps}{\partial \beta} = DPXY - \frac{1}{R_b} u - B_y$$

\*DOC.INTE

```

1  SUBROUTINE INTE (I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3
2  C,AI,AH,AV,TAUX,TAUY
3  C,DX,DY,DZ,D,E,DT,DPSX,DPSY,AP)
4  DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),MAR(IN,JN),HI(IN,JN)
5  DIMENSION HX(IN,JN),HY(IN,JN)
6  DIMENSION A3(KN)
7  DIMENSION XINT(IN,JN),YINT(IN,JN)
8  DIMENSION DPSX(IN,JN),DPSY(IN,JN)
9  DIMENSION D(IN,JN,KN),E(IN,JN,KN)
10 DO 200 I=1,IN
11 DO 200 J=1,JN
12 IF (MAR(I,J).EQ.C) GO TO 200
13 YINT(I,J)=0.0
14 XINT(I,J)=0.0
15 DO 190 K=1,KN
16 CALL D1NERU(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1HUUX,D1HUVX,MAR)
17 CALL D1UVY(I,J,K,IN,JN,KN,U,V,HI,DY,D1HUVY,MAR)
18 CALL D1VVY(I,J,K,IN,JN,KN,U,V,HI,DY,D1HVVY,MAR)
19 CALL D1VU(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1UX,D2UX,D1UY,D2UY,MAR)
20 CALL D1VSV(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1VX,D2VX,D1VY,D2VY,MAR)
21 CALL D1ZZ(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,DX,DY,DZ,D1UWZ,A3,
22 CTAUX,TAUY,D1VWZ,D1UZ,D2UZ,D1VZ,D2VZ,D1A3Z)
23 IF (K.EQ.1) GO TO 1000
24 IF (K.EQ.KN) GO TO 1010
25 XSUM=(AI*(D1HUUX+D1HUVY+HI(I,J)*D1UWZ)
26 C-AH*(D2UX*HI(I,J)+D2UY*HI(I,J))
27 C-AH*(D1UX*HX(I,J)+D1UY*HY(I,J))
28 C-AV*(1.0/HI(I,J))*(A3(K)*D2UZ+D1A3Z*D1UZ))*DZ
29 YSUM=(AI*(D1HUVX+D1HVVY+HI(I,J)*D1VWZ)
30 C-AH*(D2VX*HI(I,J)+D2VY*HI(I,J))
31 C-AH*(D1VX*HX(I,J)+D1VY*HY(I,J))
32 C-AV*(1.0/HI(I,J))*(A3(K)*D2VZ+D1A3Z*D1VZ))*DZ
33 GO TO 1100
34 1000 CCNTINUE
35 XSUM=(AI*(D1HUUX+D1HUVY+HI(I,J)*D1UWZ)
36 C-AH*(D2UX*HI(I,J)+D2UY*HI(I,J))
37 C-AH*(D1UX*HX(I,J)+D1UY*HY(I,J))
38 C-AV*(1.0/HI(I,J))*(A3(K)*D2UZ+D1A3Z*D1UZ))*DZ/2.0
39 YSUM=(AI*(D1HUVX+D1HVVY+HI(I,J)*D1VWZ)
40 C-AH*(D2VX*HI(I,J)+D2VY*HI(I,J))
41 C-AH*(D1VX*HX(I,J)+D1VY*HY(I,J))
42 C-AV*(1.0/HI(I,J))*(A3(K)*D2VZ+D1A3Z*D1VZ))*DZ/2.0
43 D1UT=(U(I,J,K)-C(I,J,K))/DT
44 D1VT=(V(I,J,K)-E(I,J,K))/DT
45 Q=2.0/DZ
46 DPSX(I,J)=(1./AP)*(1./HI(I,J))*(-XSUM*Q-HI(I,J)*D1UT)
47 DPSY(I,J)=(1./AP)*(1./HI(I,J))*(-YSUM*Q-HI(I,J)*D1VT)
48 GO TO 1100
49 1010 CCNTINUE
50 XSUM=(AI*(D1HUUX+D1HUVY+HI(I,J)*D1UWZ)
51 C-AH*(D2UX*HI(I,J)+D2UY*HI(I,J))
52 C-AH*(D1UX*HX(I,J)+D1UY*HY(I,J))
53 C-AV*(1.0/HI(I,J))*(A3(K)*D2UZ+D1A3Z*D1UZ))*DZ/2.0
54 YSUM=(AI*(D1HUVX+D1HVVY+HI(I,J)*D1VWZ)
55 C-AH*(D2VX*HI(I,J)+D2VY*HI(I,J))
56 C-AH*(D1VX*HX(I,J)+D1VY*HY(I,J))

```

```

57      C-AN*(1.0/HI(I,J))*(A3(K)*D2VZ+D1A3Z+D1VZ))*DZ/2.0
58      1100  CONTINUE
59      XINT(I,J)=XSUM+XINT(I,J)
60      YINT(I,J)=YSUM+YINT(I,J)
61      190  CONTINUE
62      200  CONTINUE
63      RETURN
64      END

```



### 9.2.28 INTEB

This subroutine is used by the far field stratified model. This subroutine is similar to the subroutine INTE with the difference that it calls the subroutine VERTDF which computes the vertical viscosity and its derivative for every grid point.

IM=DJLL(1).INTEB

```

1  SUBROUTINE INTEB(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3
2  C,AI,AH,AV,TAUX,TAUY
3  C,DX,DY,DZ,U,E,DT,DPSX,DPSY,AP,T,TREF,CONS,AVMX,AVMN)
4  DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),MAR(IN,JN),HI(IN,JN)
5  DIMENSION HX(IN,JN),HY(IN,JN),T(IN,JN,KN)
6  DIMENSION A3(KN)
7  DIMENSION XINT(IN,JN),YINT(IN,JN)
8  DIMENSION DPSX(IN,JN),DPSY(IN,JN)
9  DIMENSION D(IN,JN,KN),L(IN,JN,KN)
10 DO 200 I=1,IN
11 DO 200 J=1,JN
12 IF(MAR(I,J).EQ.C) GO TO 200
13 YINT(I,J)=J.O
14 XINT(I,J)=J.O
15 DO 190 K=1,KN
16 CALL DINEFU(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1HUUX,D1HUVX,MAR)
17 CALL DUVY(I,J,K,IN,JN,KN,U,V,HI,DY,D1HUVY,MAR)
18 CALL DVVY(I,J,K,IN,JN,KN,U,V,HI,DY,D1HVY,MAR)
19 CALL DVISU(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1UX,D2UX,D1UY,D2UY,MAR)
20 CALL DVISV(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1VX,D2VX,D1VY,D2VY,MAR)
21 CALL D12Z(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,DX,DY,DZ,D1UWZ,A3,
22 CTAUX,TAUY,D1VWZ,D1UZ,D2UZ,D1VZ,D2VZ,D1A3Z)
23 CALL VERTDF(I,J,K,IN,JN,KN,HI,AB3,D1A3Z,D1B3Z,DZ,T,A3,TREF
24 C,CONS,AVMX,AVMN)
25 A3(K)=AB3
26 IF (K.EQ.1) GO TO 1000
27 IF (K.EQ.KN) GO TO 1010
28 XSUM=(AI*(D1HUUX+D1HUVY+HI(I,J)*D1UWZ)
29 C-AH*(D2UX*HI(I,J)+D2UY*HI(I,J))
30 C-AH*(D1UX*HX(I,J)+D1UY*HY(I,J))
31 C-AV*(1.0/HI(I,J))*(A3(K)*D2UZ+D1A3Z+D1UZ))*DZ
32 YSUM=(AI*(D1HUVX+D1HVY+HI(I,J)*D1VWZ)
33 C-AH*(D2VX*HI(I,J)+D2VY*HI(I,J))
34 C-AH*(D1VX*HX(I,J)+D1VY*HY(I,J))
35 C-AV*(1.0/HI(I,J))*(A3(K)*D2VZ+D1A3Z+D1VZ))*DZ
36 GO TO 1100
37 1000 CONTINUE
38 XSUM=(AI*(D1HUUX+D1HUVY+HI(I,J)*D1UWZ)
39 C-AH*(D2UX*HI(I,J)+D2UY*HI(I,J))
40 C-AH*(D1UX*HX(I,J)+D1UY*HY(I,J))
41 C-AV*(1.0/HI(I,J))*(A3(K)*D2UZ+D1A3Z+D1UZ))*DZ/2.0
42 YSUM=(AI*(D1HUVX+D1HVY+HI(I,J)*D1VWZ)
43 C-AH*(D2VX*HI(I,J)+D2VY*HI(I,J))
44 C-AH*(D1VX*HX(I,J)+D1VY*HY(I,J))
45 C-AV*(1.0/HI(I,J))*(A3(K)*D2VZ+D1A3Z+D1VZ))*DZ/2.0
46 D1UT=(U(I,J,K)-D(I,J,K))/DT
47 D1VT=(V(I,J,K)-E(I,J,K))/DT
48 Q=2.0/DZ
49 DPSX(I,J)=(1.0/AP)*(1.0/HI(I,J))*(-XSUM*Q-HI(I,J)*D1UT)
50 DPSY(I,J)=(1.0/AP)*(1.0/HI(I,J))*(-YSUM*Q-HI(I,J)*D1VT)
51 GO TO 1100
52 1010 CONTINUE
53 XSUM=(AI*(D1HUUX+D1HUVY+HI(I,J)*D1UWZ)
54 C-AH*(D2UX*HI(I,J)+D2UY*HI(I,J))
55 C-AH*(D1UX*HX(I,J)+D1UY*HY(I,J))
56 C-AV*(1.0/HI(I,J))*(A3(K)*D2UZ+D1A3Z+D1UZ))*DZ/2.0

```

```

57      YSUM=(AI*(D1HUVX+D1HVVY+HI(I,J)*D1VWZ)
58      C-AH*(D2VX*HI(I,J)+D2VY*HI(I,J))
59      C-AH*(D1VX*HX(I,J)+D1VY*HY(I,J))
60      C-AV*(1.0/HI(I,J))*(A3(K)*D2VZ+D1A3Z*D1VZ))*DZ/2.0
61      1100 CONTINUE
62      XINT(I,J)=XSUM+XINT(I,J)
63      YINT(I,J)=YSUM+YINT(I,J)
64      190 CONTINUE
65      200 CONTINUE
66      RETURN
67      END

```

## 9.2.29 INTEMP

This program inputs discharge temperature into the model at plume discharge.

TLL and TMM are non-dimensionalized temperatures. Non dimensionalized with respect to reference temperature as shown below

$$TLL = TMM = \frac{T - T_{ref}}{T_{ref}}$$

\*DOC.INTEMP

```

1      SUBROUTINE INTEMP(I,J,K,IN,JN,KN,T,TD,TLL,TMM)
2      DIMENSION T(IN,JN,KN),TD(IN,JN,KN)
3      INM1=IN-1
4      JNM1=JN-1
5      KNM1=KN-1
6      DO 100 K=1,KN
7          T(9,1,K)=TLL
8          TD(9,1,K)=TLL
9          T(10,1,K)=TMM
10         TD(10,1,K)=TMM
11         100 CONTINUE
12     RETURN
13     END

```

## 9.2.30 OLDT

This program sets the values of temperature field at time step  $n$  equal to the temperature field at  $(n + 1)$  after all computations for time step  $n$  are completed.

JOC.OLDT

```
1  SUBROUTINE OLDT(I,J,K,IN,JN,KN,T,TP)
2  DIMENSION T(IN,JN,KN),TP(IN,JN,KN)
3  DO 10 I=1,IN
4  DO 10 J=1,JN
5  DO 10 K=1,KN
6  TP(I,J,K)=T(I,J,K)
7      10 CONTINUE
8  RETURN
9  END
```

## 9.2.31 OLDUV

This program sets the values of D and E equal to U and V respectively in order to retain values of U and V at one time step lag.



DOC.OLDUV

```

1 C*****
2 C   THIS PROGRAM SETS THE VALUES OF D AND E EQUAL TO U AND V RESPECTIVELY
3 C   IN ORDER TO RETAIN VALUES OF U AND V AT ONE TIME STEP LAG
4 C*****
5 SUBROUTINE OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
6 DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN)
7 DO 831 K=1,KN
8 DO 831 I=1,IN
9 DO 831 J=1,JN
10 D(I,J,K)=U(I,J,K)
11 E(I,J,K)=V(I,J,K)
12 831 CONTINUE
13 RETURN
14 END

```

## 9.2.32 OUTEMP

This subroutine sets the boundary conditions for temperature at the outlets. It sets near field outlet temperatures equal to those at the adjacent grid.

Eg:  $T_{IN-1} = T_{IN}$

30C.OUTEMP

```

1      SUBROUTINE OUTEMP(I,J,K,IN,JN,KN,TD)
2      DIMENSION TD(IN,JN,KN)
3      INM1=IN-1
4      JNM1=JN-1
5      KNM1=KN-1
6      DO 200 K=1,KNM1
7          DO 200 J=1,JN
8              TD(I,J,K)=TD(2,J,K)
9      200 CONTINUE
10         DO 300 K=1,KNM1
11             DO 300 J=1,JN
12                 TD(IN,J,K)=TD(INM1,J,K)
13     300 CONTINUE
14         DO 400 K=1,KNM1
15             DO 400 I=2,INM1
16                 TD(I,JN,K)=TD(I,JNM1,K)
17     400 CONTINUE
18     RETURN
19     END

```

## 9.2.33 OUTVEL

This program sets the boundary conditions at the outlets for velocity. In other words, it sets near field outlet velocities equal to those at the adjacent grid for U and V at the boundaries where there is no current. This implies that gradients normal to open boundary are equal to zero.

Eg:  $U_{IN-1} = U_{IN}$

## DOC.OUTVEL

```

1      SUBROUTINE OUTVEL (I,J,K,IN,JN,KN,H,G)
2      DIMENSION H(IN,JN,KN),G(IN,JN,KN)
3      INM1=IN-1
4      JNM1=JN-1
5      KNM1=KN-1
6      DO 200 K=1,KNM1
7      DO 200 J=1,JN
8      H(1,J,K)=H(2,J,K)
9      G(1,J,K)=G(2,J,K)
10     200 CONTINUE
11     DO 300 K=1,KNM1
12     DO 300 J=1,JN
13     H(IN,J,K)=H(INM1,J,K)
14     G(IN,J,K)=G(INM1,J,K)
15     300 CONTINUE
16     DO 400 K=1,KNM1
17     DO 400 I=2,INM1
18     H(I,JN,K)=H(I,JNM1,K)
19     G(I,JN,K)=G(I,JNM1,K)
20     400 CONTINUE
21     RETURN
22     END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

## 9.2.34 PDPSXY

This program prints  $\frac{\partial p_s}{\partial \alpha}$  and  $\frac{\partial p_s}{\partial \beta}$  at main grid points.

I=DOC.PDPSXY

```

1 SUBROUTINE PDPSXY(I,J,IN,JN,DPSX,DPSY)
2 DIMENSION DPSX(IN,JN),DPSY(IN,JN)
3 DO 10 I=1,IN
4 PRINT 1,I,(DPSX(I,J),J=1,JN)
5 PRINT 2,(DPSY(I,J),J=1,JN)
6       1 FORMAT(/' I=' ,I/' DPSX'/(5X,8E15.7))
7       2 FORMAT(' DPSY'/(5X,8E15.7))
8       10 CONTINUE
9 RETURN
10 END

```

## 9.2.35 POTUV

This program plots surface velocities for the near field.



\*DOC.POTUV

C

PLOTS U AND V ON CONSTANT DEPTH SECTIONS

PARAMETER IN=10,JN=21,IWN=17,JWN=20,KN=5,KNM1=4

DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),

CUH(IWN,JWN,KN),W(IN,JN,KN),WR(IN,JN,KN),WRH(IWN,JWN,KN),

CHI(IN,JN),HX(IN,JN),HY(IN,JN),HAR(IN,JN),MRH(IWN,JWN)

DIMENSION TW(IWN,JWN,KN),RO(IN,JN,KN),PINTH(IWN,JWN),ROW(IWN,JWN,

CKN),T(IN,JN,KN)

DIMENSION IBUF(1000)

READ 1, IRUN

READ 1,USCALE,VSCALE

ARMIN=0.04

ARMAX=0.15

IF(IRUN.EQ.0) GO TO 4

IF(IRUN.EQ.1) GO TO 5

4 CALL READ1(U,V,W,PINTH,I,J,K,IW,JW,IN,JN,KN,

CIWN,JWN,D,E,HX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,

COY,DZ,DT,TAUX,YAUY,W,WR,WRH,TTOT)

GO TO 6

5 CALL READ2(U,V,W,PINTH,I,J,K,IW,JW,IN,JN,KN,

CIWN,JWN,D,E,HX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,

COY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT,

CCB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TRFF,TO,

CTAMB,TTOT)

CONTINUE

CALL PLOTS(IBUF,1000,11)

CALL FACTOR(0.25)

FORMAT ( )

DO 10 K=1,KN

IF(K.GT.1) GO TO 20

DO 30 I=1,IN

DO 30 J=1,JN

IF (MAR(I,J).EQ.0) GO TO 35

AI=(I-1)\*1.0

AJ=(J-1)\*1.0

AAI=AI+U(I,J,K)\*USCALE

AAJ=AJ+V(I,J,K)\*VSCALE

YW=0.2\*SQRT((AAI-AI)\*\*2+(AAJ-AJ)\*\*2)

YW=AMAX1(ARMIN/0.25,AMIN1(YW,ARMAX/C.25))

CALL AROHD(AI,AJ,AAI,AAJ,YW,0.0,12)

CONTINUE

CONTINUE

GO TO 100

CONTINUE

DEPTH=(1.0/KNM1)\*(K-1)

DO 40 I=1,IN

DO 40 J=1,JN

IF(HI(I,J).GT.DEPTH) GO TO 45

GO TO 50

CONTINUE

DDZ=HI(I,J)/KNM1

LD1=(DEPTH/HI(I,J))\*KNM1

IF(LD1.EQ.0) GO TO 55

LD2=LD1+1

LD3=LD1+2

DIFF=(DEPTH-LD1\*DDZ)

COEFFS OF SECOND DEGREE FIT

ORIGINAL PAGE IS  
OF POOR QUALITY

```

57      U1=U(I,J,LD1)
58      U2=U(I,J,LD2)
59      U3=U(I,J,LD3)
60      V1=V(I,J,LD1)
61      V2=V(I,J,LD2)
62      V3=V(I,J,LD3)
63      AU=(U3-2*U2+U1)/(2*DDZ+DDZ)
64      BU=(4*U2-3*U1-U3)/(2*DDZ)
65      CU=U1
66      AV=(V3-2*V2+V1)/(2*DDZ+DDZ)
67      BV=(4*V2-3*V1-V3)/(2*DDZ)
68      CV=V1
69      AZ=DDZ+D IFF
70      UDEPTH=AU*AZ+AZ+BU*AZ+CU
71      VDEPTH=AV*AZ+AZ+BV*AZ+CV
72      GO TO 60
73      55 CONTINUE
74      AZ=DEPTH
75      AU=(U(I,J,3)-2*U(I,J,2)+U(I,J,1))/(2*DDZ+DDZ)
76      BU=(4*U(I,J,2)-3*U(I,J,1)-U(I,J,3))/(2*DDZ)
77      CU=U(I,J,1)
78      UDEPTH=AU*AZ+AZ+BU*AZ+CU
79      AV=(V(I,J,3)-2*V(I,J,2)+V(I,J,1))/(2*DDZ+DDZ)
80      BV=(4*V(I,J,2)-3*V(I,J,1)-V(I,J,3))/(2*DDZ)
81      CV=V(I,J,1)
82      VDEPTH=AV*AZ+AZ+BV*AZ+CV
83      60 CONTINUE
84      AI=(I-1)*1.0
85      AJ=(J-1)*1.0
86      AAI=AI+UDEPTH*USCALE
87      AAJ=AJ+VDEPTH*VSCALE
88      YW=0.2*SQRT((AAI-AI)**2+(AAJ-AJ)**2)
89      YW=AMAX1(ARMIN/C.25,AMIN1(YW,ARMAX/0.25))
90      CALL AROND(AI,AJ,AAI,AAJ,YW,C.0,12)
91      50 CONTINUE
92      40 CONTINUE
93      100 CONTINUE
94      A=.2*USCALE
95      B=16.0
96      C=B+A
97      CALL PLOT(B,0.0,3)
98      CALL PLOT(C,0.0,2)
99      CALL PLOT(C,-0.2,2)
100     CALL PLOT(B,-0.2,2)
101     CALL PLOT(B,0.0,2)
102     CALL PLOT(50.0,0.0,-3)
103     10 CONTINUE
104     END

```

## 9.2.36 POTUW

This program plots the vertical section velocities perpendicular to the discharge for the near-field.

\*DOC.POTUM

```

1      PARAMETER IN=18,JN=21,KN=5,IWN=17,JWN=20,KNM1=4
2      DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
3      CMH(IWN,JWN,KN),W(IN,JN,KN),WR(IN,JN,KN),WRH(IWN,JWN,KN),
4      CHI(IN,JN),HX(IN,JN),HY(IN,JN),HAR(IN,JN),MRH(IWN,JWN)
5      DIMENSION TW(IWN,JWN,KN),RO(IN,JN,KN),PINTH(IWN,JWN),ROW(IWN,JWN,
6      CKN),T(IN,JN,KN)
7      DIMENSION IBUF(1000)
8      READ 1, IRUN
9      IF(IRUN.EQ.0) GO TO 4
10     IF(IRUN.EQ.1) GO TO 5
11     4  CALL READ1(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,
12     CIWN,JWN,D,E,HX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,
13     COY,DZ,DT,TAUX,YAUY,W,WR,WRH,TTOT)
14     GO TO 6
15     5  CALL READ2(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,
16     CIWN,JWN,D,E,HX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,
17     COY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT,CB,CM,
18     CA,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,
19     CTAMB,TTOT)
20     6  CONTINUE
21     CALL PLOTS(IBUF,1000,11)
22     CALL FACTOR(0.25)
23     READ 1, USCALE,VSCALE,WSCALE,HBYL
24     ARMIN=0.04
25     ARMAX=0.15
26     1  FORMAT (I)
27     DO 10 I=1,IN
28     DO 20 J=1,JN
29     IF (MAR(I,J).LT.11) GO TO 20
30     AJ=(J-1)*1.0
31     DO 15 K=1,KNM1
32     AK=-(K-1)*HI(I,J)
33     AAJ=AJ+V(I,J,K)*VSCALE
34     AAK=AK-W(I,J,K)*WSCALE*HBYL
35     YW=0.2*SQRT((AAJ-AJ)**2+(AAK-AK)**2)
36     YW=AMAX1(ARMIN/C.25,AMIN1(YW,ARMAX/C.25))
37     CALL AROHD(AJ,AK,AAJ,AAK,YW,0.0,12)
38     15  CONTINUE
39     20  CONTINUE
40     C  DRAWS BOTTOM SURFACE
41     NN=0
42     DO 30 J=1,JN
43     IF (MAR(I,J).EQ.0) GO TO 32
44     NN=NN+1
45     IF (NN.GT.1) GO TO 33
46     AAJ=(J-1)*1.0
47     AAK=-HI(I,J)*KNM1
48     CALL PLOT(AAJ,0.0,3)
49     CALL PLOT(AAJ,AAK,2)
50     GO TO 32
51     33  CONTINUE
52     AAJ=(J-1)*1.0
53     AAK=-HI(I,J)*KNM1
54     CALL PLOT(AAJ,AAK,2)
55     JD=J
56     AJD=(JD-1)*1.0

```

```
57      32  CONTINUE
58      30  CONTINUE
59          CALL PLOT(AJD,0.0,3)
60          AAK=-MI(1,JD)*KNH1
61          CALL PLOT(AJD,AAK,2)
62          A=0.2*USCALE
63          B=16.0
64          C=B*A**3
65          CALL PLOT(B,0.0,3)
66          CALL PLOT(C,0.0,2)
67          CALL PLOT(C,-0.2,2)
68          CALL PLOT(B,-C.2,2)
69          CALL PLOT(B,0.0,2)
70          CALL PLOT(30.0,0.0,-3)
71      10  CONTINUE
72      END
```

## 9.2.37 POTVW

This program plots the vertical section velocities along canal centerline for near-field.

1000C.POTVh

```

1      C      PLOTS VELOCITIES IN J SECTIONS
2      PARAMETER IN=18,JN=21,IWN=17,JWN=2C,KN=5,KNM1=4
3      DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
4      CMH(IWN,JWN,KN),PINTH(IWN,JWN)
5      DIMENSION HX(IN,JN),HY(IN,JN),HI(IN,JN),MAR(IN,JN),MRH(IWN,JWN),
6      CM(IN,JN,KN),WR(IN,JN,KN),WRH(IN,JN,KN)
7      DIMENSION T(IN,JN,KN),RO(IN,JN,KN),TW(IWN,JWN,
8      CKN),ROW(IWN,JWN,KN)
9      DIMENSION IBUF(1000)
10     READ 1, IRUN
11     IF (IRUN.EQ.0) GO TO 4
12     IF (IRUN.EQ.1) GO TO 5
13     4      CALL READ1(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,
14     CIWN,JWN,D,E,HX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,
15     CDY,DZ,DT,TAUX,TAUY,W,WR,WRH,TTOT)
16     GO TO 6
17     5      CALL READ2(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,
18     CIWN,JWN,D,E,HX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,
19     CDY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT,CB,CW,
20     CA,B,C,EUL,T,TW,PO,ROW,TE,RREF,TREF,TO,
21     CTAMB,TTOT)
22     6      CONTINUE
23     CALL PLOTS(IBUF,1000,11)
24     CALL FACTOR(0.25)
25     READ 1, USCALE,VSCALE,WSCALE,HBYL
26     ARMIN=0.04
27     ARMAX=0.15
28     1      FORMAT (1)
29     DO 10 J=1,JN
30     DO 20 I=1,IN
31     IF (MAR(I,J).LT.11) GO TO 20
32     AI=(I-1)*1.0
33     DO 30 K=1,KNM1
34     AK=-(K-1)*HI(I,J)
35     AAI=AI+U(I,J,K)*USCALE
36     W(I,1,K)=0.0
37     AAK=AK-W(I,J,K)*WSCALE*HBYL
38     YW=0.2*SQRT((AAI-AI)**2+(AAK-AK)**2)
39     YW=AMAX1(ARMIN/C.25,AMIN1(YW,ARMAX/0.25))
40     CALL AROHD(AI,AK,AAI,AAK,YW,0.0,12)
41     30      CONTINUE
42     20      CONTINUE
43     C      DRAWS BOTTOM SURFACE
44     NN=0
45     DO 35 I=1,IN
46     IF (MAR(I,J).EQ.0) GO TO 40
47     NN=NN+1
48     IF (NN.GT.1) GO TO 33
49     AAI=(I-1)*1.0
50     CALL PLOT(AAI,0.0,3)
51     AAK=-HI(I,J)*KNM1
52     CALL PLOT(AAI,AAK,2)
53     GO TO 40
54     33      CONTINUE
55     AAI=(I-1)*1.0
56     AAK=-HI(I,J)*KNM1

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

57      CALL PLOT(AAI,AAK,2)
58      ID=1
59      AID=(ID-1)*1.0
60      40  CONTINUE
61      35  CONTINUE
62      CALL PLOT(AID,0.0,3)
63      AAK=-H(I,J)*KNM1
64      CALL PLOT(AID,AAK,2)
65      A=.2*USCALE
66      B=16.0
67      C=B*A
68      CALL PLOT(B,0.0,3)
69      CALL PLOT(C,0.0,2)
70      CALL PLOT(C,-.2,2)
71      CALL PLOT(B,-.2,2)
72      CALL PLOT(B,0.0,2)
73      CALL PLOT(50.0,C.0,-3)
74      10  CONTINUE
75      END

```



**9.2.38 PRPARA**

This subroutine prints values computed after one time step. Quantities printed are AI, AH, AV, AP, DX, DY, DZ, DT, DL2, MAXIT, EPS, OMEGA, ARBP, TAUX, TAUY and TTOT.

JOC,PRPARA

```

1  SUBROUTINE PRPARA(AI,AH,AV,AP,DX,DY,DZ,DT,DL2,MAXIT,EPS,
2  COMEGA,ARBP,TAUX,TAUY,TTOT,MAR,MRH,IN,JN,IWN,JWN)
3  DIMENSION MAR(IN,JN),MRH(IWN,JWN)
4  PRINT 1, AI,AH,AV,AP,DX,DY,DZ,DT,DL2,MAXIT,EPS,OMEGA,
5  CARBP,TAUX,TAUY,TTOT
6  1  FORMAX (/ ' AI=' ,E15.7, / ' AH=' ,E15.7, / ' AV=' ,E15.7, / ' AP=' ,E15.7,
7  C / ' DX=' ,E15.7, / ' DY=' ,E15.7, / ' DZ=' ,E15.7, / ' DT=' ,E15.7, / ' DL2=' ,
8  CE15.7, / ' MAXIT=' ,I5, / ' EPS=' ,E15.7, / ' OMEGA=' ,E15.7, / ' ARBP=' ,
9  CE15.7, / ' TAUX=' ,E15.7, / ' TAUY=' ,E15.7, / ' TTOT=' ,E15.7 / )
10  DO400J=1,JN
11  JJ=JN+1-J
12  PRINT 700,(MAR(I,JJ),I=1,IN)
13  CONTINUE
14  400  DO600JW=1,JWN
15  JJW=JWN+1-JW
16  PRINT 800,(MRH(IN,JJW),IW=1,IWN)
17  CONTINUE
18  600  FORMAT(/, ' MAR ',(3X,29I3, /))
19  700  FORMAT(/, ' MRH ',(3X,28I3, /))
20  800  RETURN
21  END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

9.2.39 PRE1

This program computes pressure for far field from Poisson equation. An iterative scheme is employed. Values are computed at half grid points.

MODUL(1),PRE1

```

1      SUBROUTINE PRE1(EPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FH,DL2,OMEGA,
2      CMRH,I,J,K,IW,JW,DX,DY,LX,IXN,JWN,ARBP)
3      DIMENSION P(IWN,JWN),FH(IWN,JWN),DPSX(IN,JN),DPSY(IN,JN)
4      DIMENSION MRH(IWN,JWN)
5      ITN=0
6      EX=0.0
7      ITN=ITN+1
8      DO 13 IWD=1,IWN
9      DO 13 JW=1,JWN
10     IW=(IWN+1)-IWD
11     I=IW
12     J=JW
13     IF (MRH(IW,JW).EQ.0) GO TO 57
14     IF (MRH(IW,JW).EQ.1) GO TO 11
15     IF (MRH(IW,JW).EQ.2) GO TO 12
16     IF (MRH(IW,JW).EQ.3) GO TO 13
17     IF (MRH(IW,JW).EQ.4) GO TO 14
18     IF (MRH(IW,JW).EQ.5) GO TO 18
19     IF (MRH(IW,JW).EQ.6) GO TO 16
20     IF (MRH(IW,JW).EQ.7) GO TO 17
21     IF (MRH(IW,JW).EQ.8) GO TO 18
22     IF (MRH(IW,JW).EQ.10) GO TO 19
23     PN=.25*(P(IW-1,JW)+P(IW+1,JW)+P(IW,JW-1)+P(IW,JW+1)-DL2*FH(IW,JW))
24     GO TO 50
25     11 CONTINUE
26     PN=.25*(P(IW-1,JW)+P(IW+1,JW)+P(IW,JW-1)+P(IW,JW+1)+(DPSY(I,J+1)
27     C+DPSY(I+1,J+1))*DY/2.-DL2*FH(IW,JW))
28     GO TO 50
29     12 CONTINUE
30     PN=.25*(P(IW-1,JW)+P(IW+1,JW)+P(IW,JW+1)+P(IW,JW)
31     C-(DPSY(I,J)+DPSY(I+1,J))*DY/2.-DL2*FH(IW,JW))
32     GO TO 50
33     13 CONTINUE
34     PN=.25*(P(IW+1,JW)+P(IW,JW-1)+P(IW,JW)-(DPSX(I,J)+DPSX(I,J+1))
35     C+DX/2*P(IW,JW)+(DPSY(I,J+1)+DPSY(I+1,J+1))*DY/2.-DL2*FH(IW,JW))
36     GO TO 50
37     14 CONTINUE
38     PN=.25*(P(IW+1,JW)+P(IW,JW+1)+P(IW,JW)-(DPSX(I,J)+DPSX(I,J+1))*
39     C+DX/2*P(IW,JW)-(DPSY(I,J)+DPSY(I+1,J))*DY/2.-DL2*FH(IW,JW))
40     GO TO 50
41     15 CONTINUE
42     PN=ARBP
43     GO TO 50
44     16 CONTINUE
45     PN=.25*(P(IW,JW+1)+P(IW-1,JW)+P(IW,JW)+(DPSX(I+1,J+1)+DPSX(I+1,J))
46     C+DX/2*P(IW,JW)-(DPSY(I,J)+DPSY(I+1,J))*DY/2.-DL2*FH(IW,JW))
47     GO TO 50
48     17 CONTINUE
49     PN=.25*(P(IW-1,JW)+P(IW,JW-1)+P(IW,JW)+(DPSX(I+1,J)+DPSX(I+1,J+1))
50     C+DX/2*P(IW,JW)+(DPSY(I,J+1)+DPSY(I+1,J+1))*DY/2.-DL2*FH(IW,JW))
51     GO TO 50
52     18 CONTINUE
53     PN=.25*(P(IW,JW+1)+P(IW-1,JW)+P(IW,JW-1)+P(IW,JW)+(DPSX(I+1,J)
54     C+DPSX(I+1,J+1))*DX/2.-DL2*FH(IW,JW))
55     GO TO 50
56     19 CONTINUE

```

```

57      PN=0.25*(P(IW,JW-1)+P(IW,JW+1)+P(IW+1,JW)+P(IW,JW+
58      C-(DPSX(I,J)+DPSX(I,J+1))*CX/2-DL2*FH(IW,JW))
59      50  CONTINUE
60      PNEW=OMEGA*PN+(1-OMEGA)*P(IW,JW)
61      IF(ABS(PNEW).LT.(10.**-16.)) GO TO 51
62      DIFF=ABS((PNEW-P(IW,JW))/PNEW)
63      IF (DIFF.LT.EX) GO TO 51
64      20  EX=DIFF
65      51  P(IW,JW)=PNEW
66      57  CONTINUE
67      10  CONTINUE
68      HT=P(1,6)-ARBP
69      DO 30 IW=1,IWN
70      DO 30 JW=1,JWN
71      IF (MRH(IW,JW).EQ.0) GO TO 31
72      P(IW,JW)=P(IW,JW)-HT
73      31  CONTINUE
74      30  CONTINUE
75      IF(EX.LT.EPS) GO TO 20
76      IF(ITN.LT.MAXIT) GO TO 1
77      20  CONTINUE
78      RETURN
79      END

```

## 9.2.40 PRE2

This program computes surface pressure for near field from Poisson equation. An iterative scheme is employed. Values are computed at half grid points.

000C.PRE2

```

1      SUBROUTINE PRE2 (EPS, MAXIT, IN, JN, P, ITN, DPSX, DPSY, FH, DL2, OMEGA,
2      1MRH, I, J, K, IW, JW, DX, DY, EX, IWN, JWN, ARBP)
3      DIMENSION P (IWN, JWN), FH (IWN, JWN), DPSX (IN, JN), DPSY (IN, JN)
4      DIMENSION MRH (IWN, JWN)
5      ITN=0
6      1    EX=0.0
7      ITN=ITN+1
8      DD=ARBP
9      DO 10 IWD=1, IWN
10     DO 10 JW=1, JWN
11     IW=(IWN+1)-IWD
12     I=IW
13     J=JW
14     IF (MRH(IW, JW).EQ.1) GO TO 2
15     IF (MRH(IW, JW).EQ.1) GO TO 11
16     IF (MRH(IW, JW).EQ.2) GO TO 12
17     IF (MRH(IW, JW).EQ.3) GO TO 13
18     IF (MRH(IW, JW).EQ.4) GO TO 14
19     IF (MRH(IW, JW).EQ.5) GO TO 18
20     IF (MRH(IW, JW).EQ.6) GO TO 16
21     IF (MRH(IW, JW).EQ.7) GO TO 17
22     IF (MRH(IW, JW).EQ.8) GO TO 18
23     IF (MRH(IW, JW).EQ.9) GO TO 19
24     IF (MRH(IW, JW).EQ.10) GO TO 20
25     2    CONTINUE
26     PN=.25*(P(IW-1, JW)+P(IW+1, JW)+P(IW, JW-1)+P(IW, JW+1)-DL2*FH(IW, JW))
27     GO TO 50
28     11    CONTINUE
29     PN=.25*(P(IW-1, JW)+P(IW+1, JW)+P(IW, JW-1)+DD-DL2*FH(IW, JW))
30     GO TO 50
31     12    CONTINUE
32     PN=.25*(P(IW-1, JW)+P(IW+1, JW)+P(IW, JW+1)+P(IW, JW)
33     C-(DPSY(I, J)+DPSY(I+1, J))*DY/2-DL2*FH(IW, JW))
34     GO TO 50
35     13    CONTINUE
36     PN=.25*(P(IW+1, JW)+P(IW, JW-1)+2*DD-DL2*FH(IW, JW))
37     GO TO 50
38     14    CONTINUE
39     PN=.25*(P(IW, JW+1)+P(IW+1, JW)+2*DD-DL2*FH(IW, JW))
40     GO TO 50
41     16    CONTINUE
42     PN=.25*(P(IW, JW+1)+P(IW-1, JW)+2*DD-DL2*FH(IW, JW))
43     GO TO 50
44     17    CONTINUE
45     PN=.25*(P(IW, JW-1)+P(IW-1, JW)+2*DD-DL2*FH(IW, JW))
46     GO TO 50
47     18    CONTINUE
48     PN=.25*(P(IW-1, JW)+P(IW, JW-1)+P(IW, JW+1)+DD-DL2*FH(IW, JW))
49     GO TO 50
50     19    CONTINUE
51     PN=.25*(P(IW-1, JW)+P(IW+1, JW)+P(IW, JW-1)+P(IW, JW+1)-DL2*FH(IW, JW))
52     GO TO 50
53     20    CONTINUE
54     PN=.25*(P(IW+1, JW)+P(IW, JW-1)+P(IW, JW+1)+DD-DL2*FH(IW, JW))
55     50    CONTINUE
56     PNEW=OMEGA*PN+(1-OMEGA)*P(IW, JW)

```

ORIGINAL FILED  
OF POOR QUALITY

```

57      IF(ABS(PNEW).LT.(10.**-16.)) GO TO 51
58      DIFF=ABS((PNEW-P(IN,JW))/PNEW)
59      IF (DIFF.LT.EX) GO TO 51
60      EX=DIFF
61      51  P(IN,JW)=PNEW
62      10  CONTINUE
63      IF(EX.LT.EPS) GO TO 32
64      IF(ITN.LT.MAXIT) GO TO 1
65      32  CONTINUE
66      RETURN
67      END

```



## 9.2.41 PREROR

This subroutine prints Hirt and Harlow correction term,  
(WHLDT)

\*DOC.PREROR

```
1 C*****
2 C   THIS PROGRAM PRINTS THE HIRT AND HARLOW CORRECTION TERM
3 C*****
4   SUBROUTINE PREROR(IWN,IV,WHLDT,JW,JWN,IN,JN)
5   DIMENSION WHLDT(IWN,JWN)
6   DO 7500 IW=1,IWN
7   7500 PRINT*7510,IW,(WHLDT(IW,JW),JW=1,JWN)
8   7510 FORMAT(/' IW=',I3,/' WHLDT'/(5X,8E15.7))
9   RETURN
10  END
```

**9.2.42 PRINTE**

This subroutine prints XINT and YINT used in computing surface pressure.

\*DOC.PRINT

```

1 C*****
2 C   THIS PROGRAM PRINTS THE INTEGRATED VALUES OF COMBINED INERTIA AND
3 C   AND VISCOUS TERMS IN U AND V MOMENTUM EQUATIONS
4 C*****
5   SUBROUTINE PRINTE (I,J,IW,JW,XINT,YINT,IN,JN)
6     DIMENSION XINT(IN,JN),YINT(IN,JN)
7     DO 8000 I=1,IN
8       PRINT 8100,I,(XINT(I,J),J=1,JN)
9       8000 PRINT 8200,(YINT(I,J),J=1,JN)
10      8100 FORMAT(/' I=',I3,/' XINT'/(5X,8E15.7))
11      8200 .FORMAT(' YINT'/(5X,8E15.7))
12      RETURN
13      END

```

9.2.43 PREPINT

This subroutine prints PINTH which is surface pressure.

\*DOC.PRINT

```

1  C*****
2  C   THIS PROGRAM PRINTS THE VALUE OF PINTH OVER THE SURFACE
3  C*****
4  SUBROUTINE PRPIAT (IW,JW,IWN,JWN,PINTH)
5  DIMENSION PINTH (IWN,JWN)
6  S200 DO 5300 IW=1,IWN
7  PRINT S400,IW,(PINTH(IW,JW),JW=1,JWN)
8  S400 FORMAT(/' IW=',I3,/(5X,8E15.7))
9  S300 CONTINUE
10 RETURN
11 END

```

**9.2.44 PRITEX**

This subroutine prints the number of iterations and final residual error in solving Poisson equation.

DOC.PRITEX

```

1 C*****
2 C   THIS PROGRAM PRINTS OUT THE VALUES OF NUMBER OF ITERATIONS AND FINAL
3 C   RESIDUAL ERROR IN SOLVING POISSON
4 C*****
5   SUBROUTINE PRITEX(ITN,EX)
6   PRINT 5500,ITN,EX
7   5500 FORMAT(1,' ITN=',I4,5X,' EX=',E15.7)
8   RETURN
9   END

```



**9.2.45 PRSORC**

This subroutine prints source term (Right hand side)  
in Poissons equation for surface pressure. (Eq. 2.17 Vol.1)

\*DOC,PRSORC

```

1 C*****
2 C   THIS PROGRAM PRINTS THE VALUE OF THE SOURCE TERM IN POISSON EQUATION
3 C*****
4   SUBROUTINE PRSORC(IM,JM,IMN,JMN,F)
5     DIMENSION F(IMN,JMN)
6     DO 6000 IM=1,IMN
7 6000 PRINT,6100,IM,(F(IM,JM),JM=1,JMN)
8 6100  FORMAT(/' IM=',I3/' SOURCE TERM'/(5X,6E15.7))
9     RETURN
10    END

```

## 9.2.46 PRUV

This subroutine prints the values of U and V at all main grid points.

I\*DOC.PRUV

```

1 C.....
2 C    THIS PROGRAM PRINTS THE VALUES OF U AND V AT GRID POINTS IN THE DOMAIN
3 C.....
4     SUBROUTINE PRUV(I,J,K,IN,JN,KN,U,V)
5     DIMENSION U(IN,JN,KN),V(IN,JN,KN)
6     DO 9100 K=1,KN
7     DO 9100 I=1,IN
8     PRINT 9000,K,I,(U(I,J,K),J=1,JN)
9     PRINT 9200,(V(I,J,K),J=1,JN)
10    9000 FORMAT(/' K=' ,I3,3X,' I=' ,I3/' U-VELOCITY'/(5X,8E15.7))
11    9200 FORMAT(' V-VELOCITY'/(5X,8E15.7))
12    RETURN
13    END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

ORIGINAL PAGE IS  
OF POOR QUALITY

## 9.2.47 PRWH

This subroutine prints the values of the vertical velocities (WH) at all half grid points.

\*DOC.PRNM

```

1 C*****
2 C   THIS PROGRAM PRINTS THE VALUS OF WH IN THE DOMAIN
3 C*****
4   SUBROUTINE PRNM (IW,JW,K,IWN,JWN,KN,WH)
5   DIMENSION WH(IWN,JWN,KN)
6   DO 7000 K=1,KN
7   DO 7000 IW=1,IWN
8   7000 PRINT 7100,K,IW,(WH(IW,JW,K),JW=1,JWN)
9   7100 FORMAT(/' K=',I3,3X,' IW=',I3,/' WH-VELOCITY'/(5X,8E15.7))
10  RETURN
11  END

```

## 9.2.48 READ2

This subroutine reads in input parameters and physical quantities stored on file designated by Unit 7. Store 2 and Read 2 correspond to each other.

## \*DOC.READ2

```

1 C*****
2 C   THIS PROGRAM READS TAPE FOR DATA I FOR THE VARIABLE DENSITY CASE
3 C*****
4 SUBROUTINE READ2(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,
5 CHX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,
6 CTAI,TAH,TAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAMB,
7 TTOT):
8   DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
9   CWH(IWN,JWN,KN),PINTH(IWN,JWN)
10  DIMENSION HX(IN,JN),HY(IN,JN),HI(IN,JN),MAR(IN,JN),MRH(IWN,JWN),
11  CW(IN,JN,KN),WR(IN,JN,KN),WRH(IWN,JWN,KN)
12  DIMENSION T(IN,JN,KN),RO(IN,JN,KN),TW(IWN,JWN,KN),ROW(IWN,JWN,KN)
13  REWIND 7
14  READ (7) (((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
15  C(((V(I,J,K),K=1,KN),J=1,JN),I=1,IN),
16  C(((D(I,J,K),K=1,KN),J=1,JN),I=1,IN),
17  C(((E(I,J,K),K=1,KN),J=1,JN),I=1,IN),
18  C(((WH(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
19  C(((W(I,J,K),K=1,KN),J=1,JN),I=1,IN),
20  C(((WR(I,J,K),K=1,KN),J=1,JN),I=1,IN),
21  C(((WRH(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
22  C((PINTH(IW,JW),JW=1,JWN),I=1,IWN)
23  C(((HI(I,J),J=1,JN),I=1,IN),((HX(I,J),J=1,JN),I=1,IN),((HY(I,J),J=
24  C1,JN),I=1,IN),((MAR(I,J),J=1,JN),I=1,IN),((MRH(IW,JW),JW=1,JWN),
25  C1,IWN),(((T(I,J,K),K=1,KN),J=1,JN),I=1,IN),
26  C(((RO(I,J,K),K=1,KN),J=1,JN),I=1,IN),
27  C(((TW(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
28  C(((ROW(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
29  CTAI,TAH,TAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAMB,
30  CAI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,TTOT
31  REWIND 7
32  RETURN
33  END

```

ORIGINAL FILED  
OF POOR QUALITY



## 9.2.49 READ2A

This is a subroutine used to read information stored by the subroutine STOR2A, for a far field unstratified model. It differs primarily from READ2 in that it does not include the matrix ROW (the densities at half grid points).

KH\*DULL(1).READ2A

```

1      C*****
2      C      THIS PROGRAM READS TAPE FOR DATA 1 FOR THE VARIABLE DENSITY CASE
3      C*****
4      SUBROUTINE READ2A(U,V,KH,PINTH,I,J,K,IX,IW,IJ,JK,IK,IWN,JWN,D,E,
5      CNX,MY,HI,MAR,MRI,AI,AN,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,TAZ,WR,RRH,
6      CTAI,TAH,TAV,AKT,CB,CW,A,B,C,LUL,T,TW,RO,TE,RREF,TREF,TO,TAMB,
7      CTTOT,ITN,CX)
8      DIMENSION U(I,J,K),V(I,J,K),D(I,J,K),E(I,J,K),
9      C(IW,IJ,JK),PINTH(IW,IJ)
10     DIMENSION CX(I,J),MY(I,J),HI(I,J),MAR(I,J),MRI(IW,JW),
11     C(IW,IJ,JK),WR(IW,IJ,JK),RRH(IW,IJ,JK)
12     DIMENSION T(I,J,K),RO(I,J,K),TE(IW,JW,KN)
13     GO TO 100
14     REWIND 7
15     100    CONTINUE
16     READ (7) ((U(I,J,K),K=1,KN),J=1,JK),I=1,IN),
17     C((V(I,J,K),K=1,KN),J=1,JK),I=1,IN),
18     C((D(I,J,K),K=1,KN),J=1,JK),I=1,IN),
19     C((E(I,J,K),K=1,KN),J=1,JK),I=1,IN),
20     C((C(IW,IJ,K),K=1,KN),J=1,JK),IW=1,IWN),
21     C((W(I,J,K),K=1,KN),J=1,JK),I=1,IN),
22     C((R(I,J,K),K=1,KN),J=1,JK),I=1,IN),
23     C((RRH(IW,IJ,K),K=1,KN),JW=1,JWN),IW=1,IWN),
24     C(PINTH(IW,IJ),JW=1,JWN),I=1,IWN)
25     C,((HI(I,J),J=1,JK),I=1,IN),((HX(I,J),J=1,JK),I=1,IN),((HY(I,J),J=
26     C1,JK),I=1,IN),((HAR(I,J),J=1,JK),I=1,IN),((MRH(IW,IJ),JW=1,JWN),
27     C(IW=1,IWN),((T(I,J,K),K=1,KN),J=1,JK),I=1,IN),
28     C((RO(I,J,K),K=1,KN),J=1,JK),I=1,IN),
29     C((TW(IW,IJ,K),K=1,KN),JW=1,JWN),IW=1,IWN),
30     C((TE(IW,IJ,K),K=1,KN),JW=1,JWN),IW=1,IWN),
31     CTAI,TAH,TAV,AKT,CB,CW,A,B,C,LUL,T,TW,RO,TW,TE,RREF,TREF,TO,TAMB,
32     CAI,AN,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,ITN,CX
33     GO TO 200
34     REWIND 7
35     200    CONTINUE
36     RETURN
37     END

```

## 9.2.50 READ2B

This subroutine reads information stored by the subroutine STOR2B for a far field stratified model. It differs from READ2 in the fact that the matrices TW (temperatures at the half grid points) and ROW (densities at the half grid points) are eliminated.

4H\*OULL(1).READ26

```

1  C*****
2  C      THIS PROGRAM READS TAPE FOR DATA 1 FOR THE VARIABLE DENSITY CASE
3  C*****
4      SUBROUTINE READ26(U,V,WH,PINTH,I,J,K,IX,JX,IN,JN,KN,IWN,JWN,D,E,
5      CHX,HY,HI,MAR,MKH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,APH,
6      CTAI,TAH,TAV,AKT,CC,CW,A,B,C,EUL,TE,RRREF,TREF,TO,TAMB,
7      CITOT,ITN,EX)
8      DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
9      CHX(IN,JN,KN),PINTH(IN,JN)
10     DIMENSION HX(IN,JN),HY(IN,JN),HI(IN,JN),MAR(IN,JN),MKH(IWN,JWN),
11     CW(IN,JN,KN),WR(IN,JN,KN),RRH(IWN,JWN,KN)
12     DIMENSION T(IN,JN,KN),RO(IN,JN,KN)
13     GO TO 100
14     REWIND 7
15     100 CONTINUE
16     READ (7) ((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
17     C((V(I,J,K),K=1,KN),J=1,JN),I=1,IN),
18     C((D(I,J,K),K=1,KN),J=1,JN),I=1,IN),
19     C((E(I,J,K),K=1,KN),J=1,JN),I=1,IN),
20     C((WH(I,J,K),K=1,KN),J=1,JN),I=1,IWN),
21     C((X(I,J,K),K=1,KN),J=1,JN),I=1,IN),
22     C((W(I,J,K),K=1,KN),J=1,JN),I=1,IN),
23     C((RRH(I,JW,K),K=1,KN),JW=1,JWN),I=1,IWN),
24     C((PINTH(IN,JN),JN=1,JWN),I=1,IWN)
25     C,((HI(I,J),J=1,JN),I=1,IN),((HX(I,J),J=1,JN),I=1,IN),((HY(I,J),J=
26     C1,JN),I=1,IN),((MAR(I,J),J=1,JN),I=1,IN),((MRH(IN,JW),JW=1,JWN),
27     C1,IWN),((T(I,J,K),K=1,KN),J=1,JN),I=1,IN),
28     C((RO(I,J,K),K=1,KN),J=1,JN),I=1,IN),
29     CTAI,TAH,TAV,AKT,CC,CW,A,B,C,EUL,TE,RRREF,TREF,TO,TAMB,
30     CTAI,TAH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,ITOT,ITN,EX
31     GO TO 200
32     REWIND 7
33     200 CONTINUE
34     RETURN
35     END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

## 9.2.51 READ3

This program classifies into interior, corner or boundary points as shown in Figs. (9.5 and 9.6).

Matrix MAR classifies points on the main grid.

MAR = 0, Point outside the region of interest.

MAR = 1, Point on the far y-boundary.

MAR = 2, Point on the near y-boundary.

MAR = 3, Point on the near x-boundary

MAR = 4, Point on the far x-boundary.

MAR = 5, Far corner on y-axis

MAR = 7, Corner at Origin

MAR = 9, Far corner on x-axis

MAR = 10, Corner at the far x-boundary and far y-boundary

Matrix MRH classifies point on the half grid.

MRH = 1, Corner at the far x-boundary and far y-boundary.

MRH = 2, Points on near y-boundary.

MRH = 3, Points on near x-boundary

MRH = 4, Corner at the near x and y boundaries

MRH = 6, Far corner on x-axis

MRH = 7, Corner at the far x and y bounds.

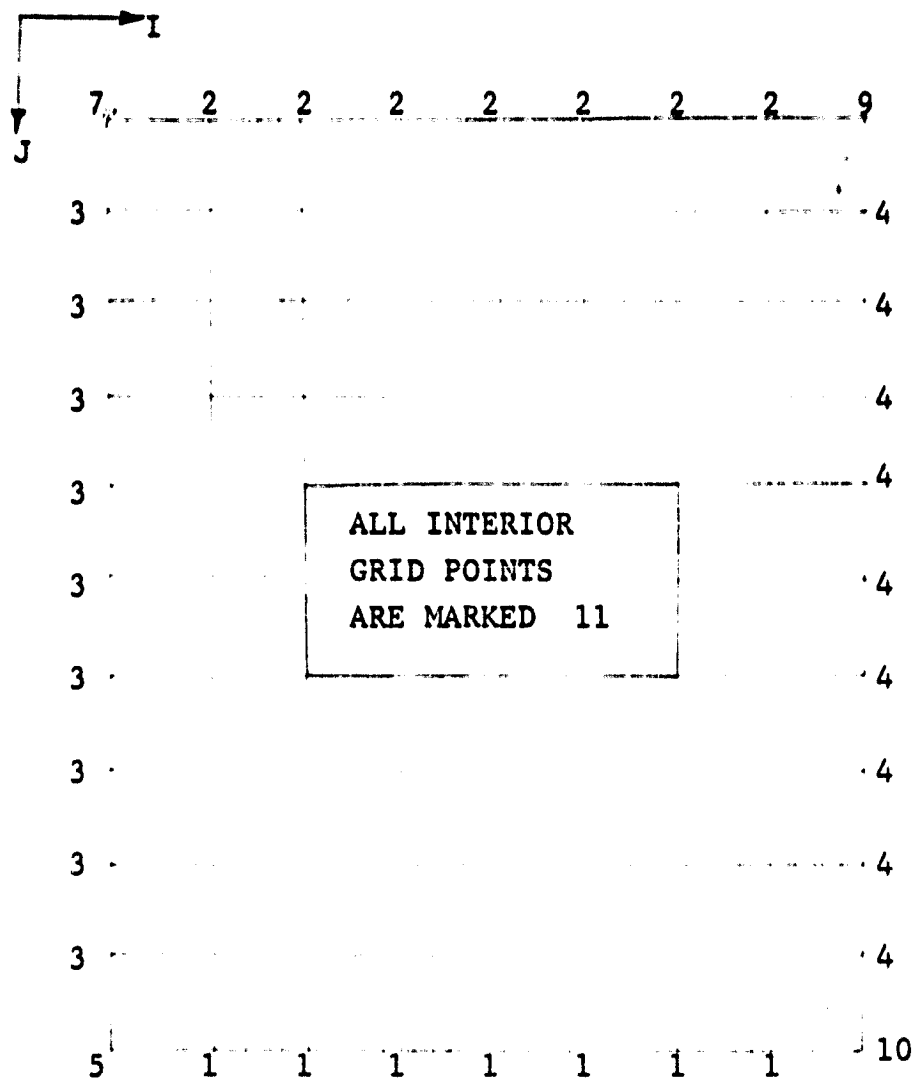


Fig: 9.5 Representation of Identifying numbers in the main grid system for near-field.

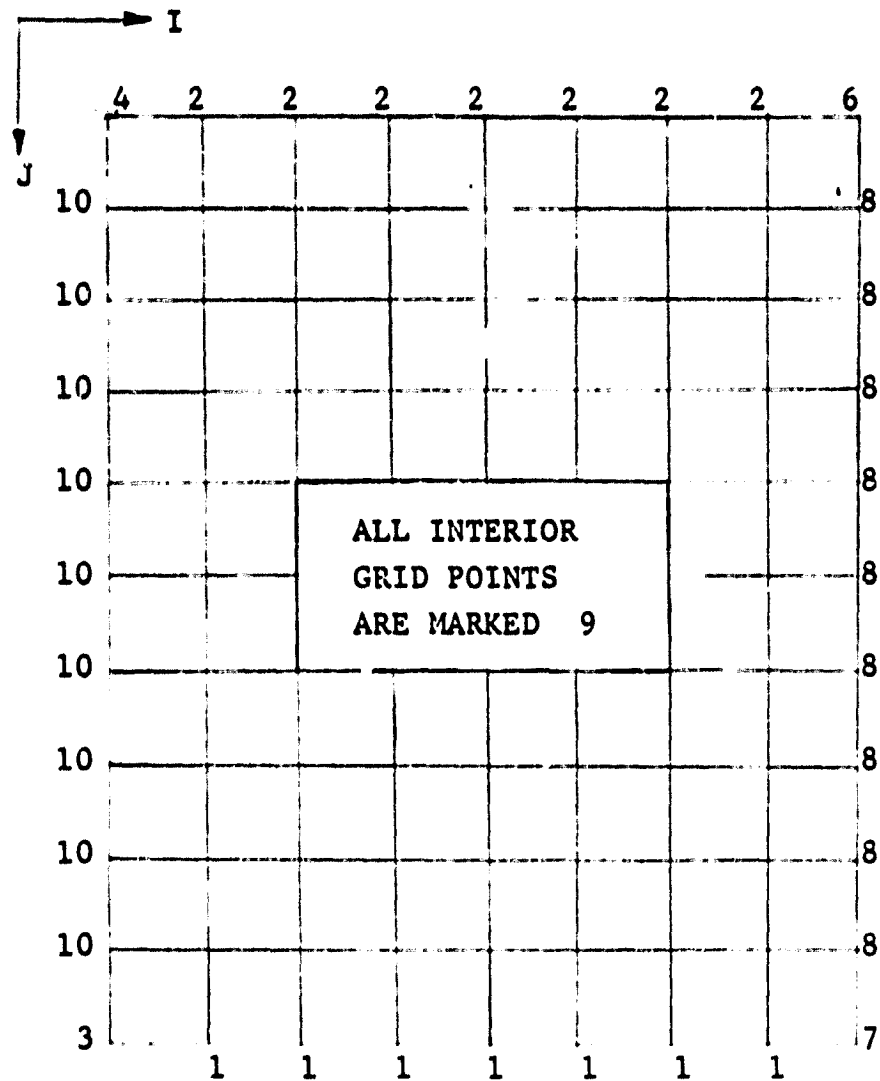


Fig:9.6 Representation of Identifying numbers in the half grid system for near-field.

1000C.READ3

```

1      SUBROUTINE READ3(I,J,IN,JN,IW,JW,IWN,JWN,MAR,MRH)
2      DIMENSION MAR(IN,JN),MRH(IWN,JWN)
3      IN1=IN-1
4      JN1=JN-1
5      IWN1=IWN-1
6      JWN1=JWN-1
7      DO 100 I=2,IN1
8      DO 100 J=2,JN1
9      MAR(I,J)=11
10     100 CONTINUE
11     DO 200 IW=2,IWN1
12     DO 200 JW=2,JWN1
13     MRH(IW,JW)=9
14     200 CONTINUE
15     DO 300 I=2,IN1
16     MAR(I,JN)=1
17     MAR(I,1)=2
18     300 CONTINUE
19     DO 400 J=2,JN1
20     MAR(1,J)=3
21     MAR(IN,J)=4
22     400 CONTINUE
23     DO 500 IW=2,IWN1
24     MRH(IW,JWN)=1
25     MRH(IW,1)=2
26     500 CONTINUE
27     DO 600 JW=2,JWN1
28     MRH(1,JW)=10
29     MRH(IWN,JW)=8
30     600 CONTINUE
31     MAR(1,1)=7
32     MAR(1,JN)=5
33     MAR(IN,JN)=10
34     40
35     MAR(IN,1)=9
36     MRH(1,1)=4
37     MRH(1,JWN)=3
38     MRH(IWN,JWN)=7
39     MRH(IWN,1)=6
40     RETURN
41     END

```



9.2.52 ROINTX

This subroutine computes  $x_p$  in the Poisson's equation (Eq. 2.17, Vol.1)  $x_p$  is then added to 'XINT'.

\*DOC,ROINTX

```

1      SUBROUTINE ROUNTX(I,J,K,IN,JN,KN,CX,CY,CZ,RO,AP,EUL,HI,
2      CHAR,RINTX,MX,XINT)
3      DIMENSION RINTX(IN,JN,KN),RO(IN,JN,KN),XINT(IN,JN),HI(IN,JN),
4      CHAR(IN,JN),HX(IN,JN)
5      DO 100 I=1,IN
6      DO 100 J=1,JN
7      IF (CHAR(I,J).EQ.0) GO TO 101
8      RINTX(I,J,1)=C.C
9      DO 110 K=2,KN
10     IF (CHAR(I,J).EQ.1) GO TO 11
11     IF (CHAR(I,J).EQ.2) GO TO 12
12     IF (CHAR(I,J).EQ.3) GO TO 13
13     IF (CHAR(I,J).EQ.4) GO TO 14
14     IF (CHAR(I,J).EQ.5) GO TO 15
15     IF (CHAR(I,J).EQ.6) GO TO 16
16     IF (CHAR(I,J).EQ.7) GO TO 17
17     IF (CHAR(I,J).EQ.8) GO TO 18
18     IF (CHAR(I,J).EQ.9) GO TO 19
19     IF (CHAR(I,J).EQ.10) GO TO 20
20     RX=DZ*(RO(I+1,J,K)+RO(I+1,J,K-1)-RO(I-1,J,K)-RO(I-1,J,K-1))/(4*DX)
21     GO TO 102
22     11 CONTINUE
23     RX=DZ*(RO(I+1,J,K)+RO(I+1,J,K-1)-RO(I-1,J,K)-RO(I-1,J,K-1))/(4*DX)
24     GO TO 102
25     12 CONTINUE
26     RX=DZ*(RO(I+1,J,K)+RO(I+1,J,K-1)-RO(I-1,J,K)-RO(I-1,J,K-1))/(4*DX)
27     GO TO 102
28     13 CONTINUE
29     RX=DZ*(4*(RO(I+1,J,K)+RO(I+1,J,K-1))-3*(RO(I,J,K)+RO(I,J,K-1))
30     C-(RO(I+2,J,K)+RO(I+2,J,K-1)))/(4*DX)
31     14 CONTINUE
32     RX=DZ*(3*(RO(I,J,K)+RO(I,J,K-1))+(RO(I-2,J,K)+RO(I-2,J,K-1))
33     C-4*(RO(I-1,J,K)+RO(I-1,J,K-1)))/(4*DX)
34     GO TO 102
35     15 CONTINUE
36     RX=DZ*(4*(RO(I+1,J,K)+RO(I+1,J,K-1))-3*(RO(I,J,K)+RO(I,J,K-1))
37     C-(RO(I+2,J,K)+RO(I+2,J,K-1)))/(4*DX)
38     GO TO 102
39     16 CONTINUE
40     RX=DZ*(RO(I+1,J,K)+RO(I+1,J,K-1)-RO(I-1,J,K)-RO(I-1,J,K-1))/(4*DX)
41     GO TO 102
42     17 CONTINUE
43     RX=DZ*(4*(RO(I+1,J,K)+RO(I+1,J,K-1))-3*(RO(I,J,K)+RO(I,J,K-1))
44     C-(RO(I+2,J,K)+RO(I+2,J,K-1)))/(4*DX)
45     GO TO 102
46     18 CONTINUE
47     RX=DZ*(RO(I+1,J,K)+RO(I+1,J,K-1)-RO(I-1,J,K)-RO(I-1,J,K-1))/(4*DX)
48     GO TO 102
49     19 CONTINUE
50     RX=DZ*(3*(RO(I,J,K)+RO(I,J,K-1))+(RO(I-2,J,K)+RO(I-2,J,K-1))
51     C-4*(RO(I-1,J,K)+RO(I-1,J,K-1)))/(4*DX)
52     GO TO 102
53     20 CONTINUE
54     RX=DZ*(3*(RO(I,J,K)+RO(I,J,K-1))+(RO(I-2,J,K)+RO(I-2,J,K-1))
55     C-4*(RO(I-1,J,K)+RO(I-1,J,K-1)))/(4*DX)
56     GO TO 102

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

57      102  CONTINUE
58      RINTX(I,J,K)=RINTX(I,J,K-1)+RX*HI(I,J)+(RO(I,J,K)+RO(I,J,K-1))*DZ+
59      CHX(I,J)/2.0
60      110  CONTINUE
61      101  CONTINUE
62      100  CONTINUE
63      DO 200 I=1,IN
64      DO 200 J=1,JN
65      IF (HAR(I,J).EQ.0) GO TO 201
66      DO 210 K=2,KN
67      RINTX(I,J,K)=RINTX(I,J,K)-(K-1)*DZ+HX(I,J)+(RO(I,J,K)+RO(I,J,K-1))
68      C/2.0
69      210  CONTINUE
70      201  CONTINUE
71      200  CONTINUE
72      DO 300 I=1,IN
73      DO 300 J=1,JN
74      IF (HAR(I,J).EQ.0) GO TO 301
75      DO 310 K=2,KN
76      RSUMX=(RINTX(I,J,K)+RINTX(I,J,K-1))*(DZ/2)+AP+EUL*HI(I,J)
77      XINT(I,J)=XINT(I,J)+RSUMX
78      310  CONTINUE
79      301  CONTINUE
80      300  CONTINUE
81      RETURN
82      END

```

9.2.53 ROINTY

This subroutine computes  $y_p$  in the Poisson's equation.  
 $y_p$  is then added to 'YINT'.

\*DOC,ROINTY

```

1  SUBROUTINE ROINTY(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,MAR,
2  CRINTY,MY,VINT)
3  DIMENSION RINTY(IN,JN,KN),RO(IN,JN,KN),VINT(IN,JN),HI(IN,JN),
4  CHY(IN,JN),MAR(IN,JN)
5  DO 100 I=1,IN
6  DO 100 J=1,JN
7  IF (MAR(I,J).EQ.0) GO TO 101
8  RINTY(I,J,1)=C.C
9  DO 110 K=2,KN
10 IF (MAR(I,J).EQ.1) GO TO 11
11 IF (MAR(I,J).EQ.2) GO TO 12
12 IF (MAR(I,J).EQ.3) GO TO 13
13 IF (MAR(I,J).EQ.4) GO TO 14
14 IF (MAR(I,J).EQ.5) GO TO 15
15 IF (MAR(I,J).EQ.6) GO TO 16
16 IF (MAR(I,J).EQ.7) GO TO 17
17 IF (MAR(I,J).EQ.8) GO TO 18
18 IF (MAR(I,J).EQ.9) GO TO 19
19 IF (MAR(I,J).EQ.10) GO TO 20
20 RY=DZ*(RO(I,J+1,K)+RO(I,J+1,K-1)-RO(I,J-1,K)-RO(I,J-1,K-1))/(4*DY)
21 GO TO 102
22 11 CONTINUE
23 RY=DZ*(3*(RO(I,J,K)+RO(I,J,K-1))+RO(I,J-2,K)+RO(I,J-2,K-1))
24 C-4*(RO(I,J-1,K)+RO(I,J-1,K-1)))/(4*DY)
25 GO TO 102
26 12 CONTINUE
27 RY=DZ*(4*(RO(I,J+1,K)+RO(I,J+1,K-1))-3*(RO(I,J,K)+RO(I,J,K-1))
28 C-(RO(I,J+2,K)+RO(I,J+2,K-1)))/(4*DY)
29 GO TO 102
30 13 CONTINUE
31 RY=DZ*(RO(I,J+1,K)+RO(I,J+1,K-1)-RO(I,J-1,K)-RO(I,J-1,K-1))/(4*DY)
32 GO TO 102
33 14 CONTINUE
34 RY=DZ*(RO(I,J+1,K)+RO(I,J+1,K-1)-RO(I,J-1,K)-RO(I,J-1,K-1))/(4*DY)
35 GO TO 102
36 15 CONTINUE
37 RY=DZ*(3*(RO(I,J,K)+RO(I,J,K-1))+RO(I,J-2,K)+RO(I,J-2,K-1))
38 C-4*(RO(I,J-1,K)+RO(I,J-1,K-1)))/(4*DY)
39 GO TO 102
40 16 CONTINUE
41 RY=DZ*(RO(I,J+1,K)+RO(I,J+1,K-1)-RO(I,J-1,K)-RO(I,J-1,K-1))/(4*DY)
42 GO TO 102
43 17 CONTINUE
44 RY=DZ*(4*(RO(I,J+1,K)+RO(I,J+1,K-1))-3*(RO(I,J,K)+RO(I,J,K-1))
45 C-(RO(I,J+2,K)+RO(I,J+2,K-1)))/(4*DY)
46 GO TO 102
47 18 CONTINUE
48 RY=DZ*(RO(I,J+1,K)+RO(I,J+1,K-1)-RO(I,J-1,K)-RO(I,J-1,K-1))/(4*DY)
49 GO TO 102
50 19 CONTINUE
51 RY=DZ*(4*(RO(I,J+1,K)+RO(I,J+1,K-1))-3*(RO(I,J,K)+RO(I,J,K-1))
52 C-(RO(I,J+2,K)+RO(I,J+2,K-1)))/(4*DY)
53 GO TO 102
54 20 CONTINUE
55 RY=DZ*(3*(RO(I,J,K)+RO(I,J,K-1))+RO(I,J-2,K)+RO(I,J-2,K-1))
56 C-4*(RO(I,J-1,K)+RO(I,J-1,K-1)))/(4*DY)

```

```

57      GO TO 102
58      102  CONTINUE
59      RINTY(I,J,K)=RINTY(I,J,K-1)+RY*HI(I,J)+(RO(I,J,K)+RO(I,J,K-1))*DZ+
60      CHY(I,J)/2.0
61      110  CONTINUE
62      101  CONTINUE
63      100  CONTINUE
64      DO 200 I=1,IN
65      DO 200 J=1,JN
66      IF (MAR(I,J).EQ.0) GO TO 201
67      DO 210 K=2,KN
68      RINTY(I,J,K)=RINTY(I,J,K-1)*DZ+HY(I,J)*(RO(I,J,K)+RO(I,J,K-1))
69      C/2.0
70      210  CONTINUE
71      201  CONTINUE
72      200  CONTINUE
73      DO 300 I=1,IN
74      DO 300 J=1,JN
75      IF (MAR(I,J).EQ.0) GO TO 301
76      DO 310 K=2,KN
77      RSUMY=(RINTY(I,J,K)+RINTY(I,J,K-1))*(DZ/2)*AP+EUL*HI(I,J)
78      YINT(I,J)=YINT(I,J)+RSUMY
79      310  CONTINUE
80      301  CONTINUE
81      300  CONTINUE
82      RETURN
83      END

```

346  
 INTENTIONALLY BLANK

## 9.2.54 RWH

This subroutine uses continuity equation to compute vertical velocities at half grid points.

\*DOC.RWH

```

1      SUBROUTINE RWH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,DX,DY,DZ,
2      C*RH)
3      DIMENSION U(IW,JN,KN),V(IN,JN,KN),WH(IWN,JN,KN),HI(IN,JN)
4      DIMENSION MRH(IW,JW)
5      KNMI=KN-1
6      DO 10 IW=1,IWN
7      DO 10 JW=1,JWN
8      IF (MRH(IW,JW).EQ.0) GO TO 8
9      DO 9 KD=1,KNMI
10     K=KN-KD+1
11     I=IW
12     J=JW
13     D1HUX=(HI(I+1,J+1)*(U(I+1,J+1,K)+U(I+1,J+1,K-1))+HI(I+1,J)*
14     C(U(I+1,J,K)+U(I+1,J,K-1))-HI(I,J+1)*(U(I,J+1,K)+U(I,J+1,K-1))
15     C-HI(I,J)*(U(I,J,K)+U(I,J,K-1)))/(4*DX)
16     D1HVV=(HI(I+1,J+1)*(V(I+1,J+1,K)+V(I+1,J+1,K-1))+HI(I,J+1)*
17     C(V(I+1,J+1,K)+V(I+1,J+1,K-1))-HI(I+1,J)*(V(I+1,J,K)+V(I+1,J,K-1))
18     C-HI(I,J)*(V(I,J,K)+V(I,J,K-1)))/(4*DY)
19     HH=(HI(I+1,J+1)+HI(I+1,J)+HI(I,J+1)+HI(I,J))/4.0
20     WH(IW,JW,K-1)=WH(IW,JW,K)+(1.0/HH)*(D1HUX+D1HVV)*DZ
21     CONTINUE
22     8    CONTINUE
23     10   CONTINUE
24     RETURN
25     END

```



**9.2.55 RWR**

Computes real vertical velocities from modified vertical velocities used in equations at integral grid points.

\*DOC.RVR

```

1      SUBROUTINE RVR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,MY,DZ,PAR)
2      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WR(IN,JN,KN),
3      CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN)
4      DO 1C I=1,IN
5      DO 1C J=1,JN
6      IF (MAR(I,J).LT.11) GO TO 8
7      KNH1=KN-1
8      DO 9 K=1,KNH1
9      WR(I,J,K)=(K-1)*DZ*(U(I,J,K)+HX(I,J)+V(I,J,K)+HY(I,J))+HI(I,J)
10     C*V(I,J,K)
11     CONTINUE
12     CONTINUE
13     CONTINUE
14     RETURN
15     END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

## 9.2.56 RWRH

Computes real vertical velocities at half grid points.

\*DOC.RURH

```

1      SUBROUTINE RURH (I,J,K,IU,JU,IN,JN,KN,IUN,JUN,U,V,MH,HI,HX,MY,
2      COX,DY,DZ,MRR,VRH)
3      DIMENSION U(IN,JN,KN),V(IN,JN,KN),MH(IUN,JUN,KN),HI(IN,JN),
4      CHX(IN,JN),HY(IN,JN),MRR(IUN,JUN)
5      DIMENSION VRH(IUN,JUN,KN)
6      KKN=KN-1
7      DO 10 IU=1,IUN
8      DO 10 JU=1,JUN
9      IF (MRR(IU,JU).EQ.0) GO TO 8
10     HXAV=(HX(I+1,J)+HX(I+1,J+1)+HX(I,J)+HX(I,J+1))/4.
11     HYAV=(HY(I+1,J)+HY(I+1,J+1)+HY(I,J)+HY(I,J+1))/4.
12     HIAV=(HI(I+1,J)+HI(I+1,J+1)+HI(I,J)+HI(I,J+1))/4.
13     DO 9 K=1,KKN
14     I=IU
15     J=JU
16     UAV=(U(I+1,J,K)+U(I+1,J+1,K)+U(I,J,K)+U(I,J+1,K))/4.
17     VAV=(V(I+1,J,K)+V(I+1,J+1,K)+V(I,J,K)+V(I,J+1,K))/4.
18     VRH(IU,JU,K)=(K-1)*DZ*(UAV*HXAV+VAV*HYAV)+HIAV*MH(IU,JU,K)
19     CONTINUE
20     8    CONTINUE
21     1C   CONTINUE
22     RETURN
23     END

```

**9.2.57 STORE2**

This subroutine stores values of input parameters and physical quantities on a file designated as Unit 8.

## DOC STORE 2

```

1  C*****
2  C      THIS PROGRAM STORES THE RELEVANT DATA INTO FILE 8
3  C*****
4  SUBROUTINE STORE2(U,V,WH,PINTH,I,J,K,IN,JN,KN,IWN,JWN,D,E,
5  CHX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,
6  CTAI,TAH,TAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAMB,
7  CTTOT)
8  DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
9  CWH(IWN,JWN,KN),PINTH(IWN,JWN)
10 DIMENSION HX(IN,JN),HY(IN,JN),HI(IN,JN),MAR(IN,JN),MRH(IWN,JWN),
11 CW(IN,JN,KN),WR(IN,JN,KN),WRH(IWN,JWN,KN)
12 DIMENSION T(IN,JN,KN),RO(IN,JN,KN),TW(IWN,JWN,KN),ROW(IWN,JWN,KN)
13 REWIND 8
14 WRITE (8) (((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
15 C(((V(I,J,K),K=1,KN),J=1,JN),I=1,IN),
16 C(((D(I,J,K),K=1,KN),J=1,JN),I=1,IN),
17 C(((E(I,J,K),K=1,KN),J=1,JN),I=1,IN),
18 C(((WH(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
19 C(((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
20 C(((WR(I,J,K),K=1,KN),J=1,JN),I=1,IN),
21 C(((MRH(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
22 C((PINTH(IW,JW),JW=1,JWN),IW=1,IWN)
23 C(((HI(I,J),J=1,JN),I=1,IN),((HX(I,J),J=1,JN),I=1,IN),((HY(I,J),J=
24 C1,JN),I=1,IN),((MAR(I,J),J=1,JN),I=1,IN),((MRH(IW,JW),JW=1,JWN),
25 C1W=1,IWN),(((T(I,J,K),K=1,KN),J=1,JN),I=1,IN),
26 C(((RO(I,J,K),K=1,KN),J=1,JN),I=1,IN),
27 C(((TW(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
28 C(((ROW(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
29 CTAI,TAH,TAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAMB,
30 CAI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,TTOT
31 REWIND 8
32 RETURN
33 END

```

ORIGINAL PAGE 1  
OF POOR QUALITY

## 9.2.58 TEMB2

This program makes computations for temperatures at the boundary points in the domain of interest. Central difference schemes are used for computing derivative of temperature with respect to  $\alpha$  and  $\beta$ . Heat flux specified at the vertical walls is used to compute the derivatives normal to the vertical walls. Heat flux boundary condition is used in a manner explained in TEMI4.

N\*DOC.TEMB2

```

1      SUBROUTINE TEMB2(I,J,K,IN,JN,KN,TD,CX,DY,DZ,MAR,CP,HI,AKT,CW,
2      CTAMB,HX,HY,T,TREF,TAV,TAI,TAH,BS,DT)
3      DIMENSION T(IN,JN,KN),TD(IN,JN,KN),PAR(IN,JN),HX(IN,JN),HY(IN,JN),
4      CHI(IN,JN)
5      KNM1=KN-1
6      DO 100 M=1,KN
7      DO 100 I=1,IN
8      DO 100 J=1,JN
9      D1HTUX=0.0
10     D1HTVY=0.0
11     D1TWZ=0.0
12     IF (MAR(I,J).EQ.0) GO TO 300
13     IF (MAR(I,J).EQ.11) GO TO 300
14     IF (MAR(I,J).EQ.1) GO TO 11
15     IF (MAR(I,J).EQ.2) GO TO 12
16     IF (MAR(I,J).EQ.3) GO TO 13
17     IF (MAR(I,J).EQ.4) GO TO 14
18     IF (MAR(I,J).EQ.5) GO TO 15
19     IF (MAR(I,J).EQ.6) GO TO 16
20     IF (MAR(I,J).EQ.7) GO TO 17
21     IF (MAR(I,J).EQ.8) GO TO 18
22     IF (MAR(I,J).EQ.9) GO TO 19
23     IF (MAR(I,J).EQ.10) GO TO 20
24     11 CONTINUE
25     D1TX=(T(I+1,J,K)-T(I-1,J,K))/(2*DX)
26     D2TX=(T(I+1,J,K)+T(I-1,J,K)-2*T(I,J,K))/(DX*DX)
27     D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
28     D1TY=0.0
29     D2TY=2*(T(I,I-1,K)-T(I,J,K))/(DY*DY)
30     IF (K.EQ.1) GO TO 110
31     IF (K.EQ.KN) GO TO 120
32     GO TO 200
33     12 CONTINUE
34     D1TX=(T(I+1,J,K)-T(I-1,J,K))/(2*DX)
35     D2TX=(T(I+1,J,K)+T(I-1,J,K)-2*T(I,J,K))/(DX*DX)
36     D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
37     D1TY=0.0
38     D2TY=2*(T(I,J+1,K)-T(I,J,K))/(DY*DY)
39     IF (K.EQ.1) GO TO 110
40     IF (K.EQ.KN) GO TO 120
41     GO TO 200
42     13 CONTINUE
43     D1TX=0.0
44     D2TX=2*(T(I+1,J,K)-T(I,J,K))/(DX*DX)
45     D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
46     D1TY=(T(I,J+1,K)-T(I,J-1,K))/(2*DY)
47     D2TY=(T(I,J+1,K)+T(I,J-1,K)-2*T(I,J,K))/(DY*DY)
48     IF (K.EQ.1) GO TO 110
49     IF (K.EQ.KN) GO TO 120
50     GO TO 200
51     14 CONTINUE
52     D1TX=0.0
53     D2TX=2*(T(I-1,J,K)-T(I,J,K))/(DX*DX)
54     D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
55     D1TY=(T(I,J+1,K)-T(I,J-1,K))/(2*DY)
56     D2TY=(T(I,J+1,K)+T(I,J-1,K)-2*T(I,J,K))/(DY*DY)

```

OF POOR QUALITY



```

57      IF (K.EQ.1) GO TO 110
58      IF (K.EQ.KN) GO TO 120
59      GO TO 200
60      15  CONTINUE
61          D1TX=0.0
62          D2TX=2*(T(I+1,J,K)-T(I,J,K))/(DX*DX)
63          D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
64          D1TY=0.0
65          D2TY=2*(T(I,J-1,K)-T(I,J,K))/(DY*DY)
66          IF (K.EQ.1) GO TO 110
67          IF (K.EQ.KN) GO TO 120
68          GO TO 200
69      17  CONTINUE
70          D1TX=0.0
71          D2TX=2*(T(I+1,J,K)-T(I,J,K))/(DX*DX)
72          D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
73          D1TY=0.0
74          D2TY=2*(T(I,J+1,K)-T(I,J,K))/(DY*DY)
75          IF (K.EQ.1) GO TO 110
76          IF (K.EQ.KN) GO TO 120
77          GO TO 200
78      19  CONTINUE
79          D1TX=0.0
80          D2TX=2*(T(I-1,J,K)-T(I,J,K))/(DX*DX)
81          D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
82          D1TY=0.0
83          D2TY=2*(T(I,J+1,K)-T(I,J,K))/(DY*DY)
84          IF (K.EQ.1) GO TO 110
85          IF (K.EQ.KN) GO TO 120
86          GO TO 200
87      20  CONTINUE
88          D1TX=0.0
89          D2TX=2*(T(I-1,J,K)-T(I,J,K))/(DX*DX)
90          D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
91          D1TY=0.0
92          D2TY=2*(T(I,J-1,K)-T(I,J,K))/(DY*DY)
93          IF (K.EQ.1) GO TO 110
94          IF (K.EQ.KN) GO TO 120
95          GO TO 200
96      110 CONTINUE
97          CT=AKT*((T(I,J,1)*TREF+TREF)-TAMB)/TREF
98          CT=CT*HI(I,J)
99          D2TZ=2*(T(I,J,2)-CT*DZ-T(I,J,1))/(DZ*DZ)
100         GO TO 200
101      120 CONTINUE
102          D2TZ=2*(T(I,J,K-1)-T(I,J,K))/(DZ*DZ)
103          GO TO 200
104      200 CONTINUE
105          TD(I,J,K)=(1.C/HI(I,J))*(-TAJ*(D1HTUX+D1HTVY+HI(I,J)*D1TWZ)+TAH
106          C*(D1TX*HX(I,J)+D2TX*HI(I,J)+D1TY*HY(I,J)+D2TY*HI(I,J))+TAV/HI(I,J
107          C))*E3*D2TZ)*DT*T(I,J,K)
108      16  CONTINUE
109      18  CONTINUE
110      300 CONTINUE
111      100 CONTINUE
112      RETURN
113      END

```

## 9.2.59 TEMB2A

This subroutine is used by the far field stratified model. This subroutine is similar to TEMB2 with the difference that this subroutine calls the subroutine VERTDF which supplies the value and the derivative of vertical diffusivity for every grid point.

SKH\*DULL(1).TMBZA

```

1      SUBROUTINE TEMPC(I,J,K,IN,KN,KN,TC,DX,DY,DZ,HAR,CF,HI,ALT,CW,
2      CTAMS,HX,HY,T,TREF,TAV,TAI,TAN,B3,DT)
3      DIMENSION T(IN,KN,KN),TD(IN,KN,KN),MAR(IN,KN),HX(IN,KN),HY(IN,KN),
4      CHI(IN,KN)
5      KNH1=KN-1
6      DO 100 K=1,KN
7      DO 100 I=1,IN
8      DO 100 J=1,KN
9      DHTUX=0.0
10     DHTVY=0.0
11     DHTWZ=0.0
12     IF (MAR(I,J).EQ.0) GO TO 300
13     IF (MAR(I,J).EQ.1) GO TO 300
14     IF (MAR(I,J).EQ.2) GO TO 11
15     IF (MAR(I,J).EQ.3) GO TO 12
16     IF (MAR(I,J).EQ.4) GO TO 13
17     IF (MAR(I,J).EQ.5) GO TO 14
18     IF (MAR(I,J).EQ.6) GO TO 15
19     IF (MAR(I,J).EQ.7) GO TO 16
20     IF (MAR(I,J).EQ.8) GO TO 17
21     IF (MAR(I,J).EQ.9) GO TO 18
22     IF (MAR(I,J).EQ.10) GO TO 19
23     IF (MAR(I,J).EQ.11) GO TO 20
24     11 CONTINUE
25     DTX=(T(I+1,J,K)-T(I-1,J,K))/(2*DX)
26     D2TX=(T(I+1,J,K)+T(I-1,J,K)-2*T(I,J,K))/(DX*DX)
27     D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
28     DTY=0.0
29     D2TY=2*(T(I,J+1,K)-T(I,J,K))/(DY*DY)
30     IF (K.EQ.1) GO TO 110
31     IF (K.EQ.KN) GO TO 120
32     GO TO 200
33     12 CONTINUE
34     DTX=(T(I+1,J,K)-T(I-1,J,K))/(2*DX)
35     D2TX=(T(I+1,J,K)+T(I-1,J,K)-2*T(I,J,K))/(DX*DX)
36     D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
37     DTY=0.0
38     D2TY=2*(T(I,J+1,K)-T(I,J,K))/(DY*DY)
39     IF (K.EQ.1) GO TO 110
40     IF (K.EQ.KN) GO TO 120
41     GO TO 200
42     13 CONTINUE
43     DTX=0.0
44     D2TX=2*(T(I+1,J,K)-T(I,J,K))/(DX*DX)
45     D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
46     DTY=(T(I,J+1,K)-T(I,J-1,K))/(2*DY)
47     D2TY=(T(I,J+1,K)+T(I,J-1,K)-2*T(I,J,K))/(DY*DY)
48     IF (K.EQ.1) GO TO 110
49     IF (K.EQ.KN) GO TO 120
50     GO TO 200
51     14 CONTINUE
52     DTX=0.0
53     D2TX=2*(T(I-1,J,K)-T(I,J,K))/(DX*DX)
54     D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
55     DTY=(T(I,J+1,K)-T(I,J-1,K))/(2*DY)
56     D2TY=(T(I,J+1,K)+T(I,J-1,K)-2*T(I,J,K))/(DY*DY)

```

```

57      IF (K.EQ.1) GO TO 110
58      IF (K.EQ.KN) GO TO 120
59      GO TO 200
60      15 CONTINUE
61      D1TX=0.0
62      D2TX=2*(T(I+1,J,K)-T(I,J,K))/(DX*DX)
63      D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
64      D1TY=0.0
65      D2TY=2*(T(I,J-1,K)-T(I,J,K))/(DZ*DZ)
66      IF (K.EQ.1) GO TO 110
67      IF (K.EQ.KN) GO TO 120
68      GO TO 200
69      17 CONTINUE
70      D1TX=0.0
71      D2TX=2*(T(I+1,J,K)-T(I,J,K))/(DX*DX)
72      D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
73      D1TY=0.0
74      D2TY=2*(T(I,J+1,K)-T(I,J,K))/(DY*DY)
75      IF (K.EQ.1) GO TO 110
76      IF (K.EQ.KN) GO TO 120
77      GO TO 200
78      19 CONTINUE
79      D1TX=0.0
80      D2TX=2*(T(I-1,J,K)-T(I,J,K))/(DX*DX)
81      D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
82      D1TY=0.0
83      D2TY=2*(T(I,J+1,K)-T(I,J,K))/(DY*DY)
84      IF (K.EQ.1) GO TO 110
85      IF (K.EQ.KN) GO TO 120
86      GO TO 200
87      20 CONTINUE
88      D1TX=0.0
89      D2TX=2*(T(I-1,J,K)-T(I,J,K))/(DX*DX)
90      D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
91      D1TY=0.0
92      D2TY=2*(T(I,J-1,K)-T(I,J,K))/(DZ*DZ)
93      IF (K.EQ.1) GO TO 110
94      IF (K.EQ.KN) GO TO 120
95      GO TO 200
96      110 CONTINUE
97      CT=AKT*((T(I,J,1)*TREF+TREF)-TAMB)/TREF
98      CT=CT*HI(I,J)
99      D2TZ=2*(T(I,J,2)-CT*DZ-T(I,J,1))/(DZ*DZ)
100     GO TO 200
101     120 CONTINUE
102     D2TZ=2*(T(I,J,K-1)-T(I,J,K))/(DZ*DZ)
103     GO TO 200
104     200 CONTINUE
105     TD(I,J,K)=(1.0/HI(I,J))*(-TAI*(D1HTUX+D1HTVY+HI(I,J)*D1TZ)+TAH
106     C*(D1TX+HX(I,J)+D1TX*HI(I,J)+D1TY*HY(I,J)+D2TY*HI(I,J))+TAV/HI(I,J)
107     C))*83*D2TZ)*DT+T(I,J,K)
108     16 CONTINUE
109     18 CONTINUE
110     200 CONTINUE
111     100 CONTINUE
112     RETURN
113     END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

PRECEDING PAGE BLANK NOT FILMED

## 9.2.60 TEMI4

This program computes temperatures at the interior points of the domain of interest. Computations are made at the full grid points. Central-difference schemes are used to write the first and second derivatives with respect to  $\alpha$  and  $\beta$ . Single sided difference schemes are used for derivatives with respect to  $\gamma$  at top surface and at bottom. Corner points designated by MAR= 6 and 8 are treated as interior points. Heat flux through the surface is used in specifying the first and second derivatives of temperature with respect to  $\gamma$ . Heat flux through the bottom is used in specifying the first and second derivatives of temperature with respect to  $\gamma$ . Energy equation is used to compute temperatures.

1000C.TEM14

```

1      SUBROUTINE TEM14(I,J,K,IN,JN,KN,U,V,T,TD,DX,
2      CCB,
3      CDY,DZ,B,DT,TAT,TAM,TAV,B3,HI,HX,HY,MAR,AKT,TREF,TAMR)
4      DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),HX(IN,JN),HY(IN,JN),
5      CHAR(IN,JN),T(IN,JN,KN),TD(IN,JN,KN),W(IN,JN,KN)
6      KNM1=KN-1
7      DO 10 I=1,IN
8      DO 10 J=1,JN
9      IF (MAR(I,J).EQ.6) GO TO 100
10     IF (MAR(I,J).EQ.8) GO TO 100
11     IF (MAR(I,J).LT.11) GO TO 9
12     100 CONTINUE
13     DO 8 K=1,KN
14     D1HTUX=(U(I+1,J,K)+T(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)+T(I-1,J,K)
15     C*HI(I-1,J))/(2*DX)
16     D1HTVY=(V(I,J+1,K)+T(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)+T(I,J-1,K)*
17     CHI(I,J-1))/(2*DY)
18     D1TX=(T(I+1,J,K)-T(I-1,J,K))/(2*DX)
19     D1TY=(T(I,J+1,K)-T(I,J-1,K))/(2*DY)
20     D1TWZ=(T(I,J,K+1)*W(I,J,K+1)-T(I,J,K-1)*W(I,J,K-1))/(2*DZ)
21     D2TX=(T(I+1,J,K)+T(I-1,J,K)-2*T(I,J,K))/(DX*DX)
22     D2TY=(T(I,J+1,K)+T(I,J-1,K)-2*T(I,J,K))/(DY*DY)
23     D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
24     IF (K.EQ.1) GO TO 24
25     IF (K.EQ.KN) GO TO 20
26     GO TO 21
27     20 CONTINUE
28     D1TWZ=0.0
29     D2TZ=2*(T(I,J,K-1)-T(I,J,K)+CB*HI(I,J)*DZ)/(DZ*DZ)
30     GO TO 21
31     24 CONTINUE
32     CT=AKT*((T(I,J,1)+TREF+TREF)-TAMR)/TREF
33     CT=CT*HI(I,J)
34     D1TWZ=0.0
35     D2TZ=2*(T(I,J,2)-CT*DZ-T(I,J,1))/(DZ*DZ)
36     21 CONTINUE
37     TC(I,J,K)=(1.C/HI(I,J))*(-TAT*(D1HTUX+D1HTVY+HI(I,J)*D1TWZ)
38     C*TAM*(D1TX*HX(I,J)+D2TX*HI(I,J)+D1TY*HY(I,J)+D2TY*HI(I,J))
39     C*(IAV/HI(I,J))*B3*D2TZ)+DT*T(I,J,K)
40     8 CONTINUE
41     9 CONTINUE
42     10 CONTINUE
43     RETURN
44     END

```

## 9.2.61 TEMI4B

This subroutine is used by the far field stratified model. The subroutine is similar to TEMI4 with the difference that it calls for subroutine VERTDF which supplies the value of vertical diffusivity and its derivative.

\*DJLL(1).TEMP40

```

1      SUBROUTINE TEMP40(I,J,K,IN,JN,KN,U,V,T,TD,DX,
2      CCB,
3      CDY,DZ,W,DT,TAI,TAH,TAV,B3,HZ,HX,HY,MAR,AKT,TREF,TAMB,A3
4      C,CONS,AVMX,AVMY)
5      DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),HX(IN,JN),HY(IN,JN),
6      CHAR(IN,JN),T(IN,JN,KN),TD(IN,JN,KN),W(IN,JN,KN),A3(KN)
7      KNH=KN-1
8      DO 10 I=1,IN
9      DO 10 J=1,JN
10     IF(MAR(I,J).EQ.6) GO TO 100
11     IF(MAR(I,J).EQ.8) GO TO 100
12     IF (MAR(I,J).LT.11) GO TO 9
13     100 CONTINUE
14     DO 8 K=1,KN
15     CALL VERTCF(I,J,K,IN,JN,KN,HZ,AB3,D1A3Z,D1B3Z,DZ,T,A3,TREF
16     C,CONS,AVMX,AVMY)
17     B3=AB3
18     D1HTUX=(U(I+1,J,K)*T(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)*T(I-1,J,K)
19     C*HI(I-1,J))/(2*DX)
20     D1HTVY=(V(I,J+1,K)*T(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*T(I,J-1,K)*
21     CHI(I,J-1))/(2*DY)
22     D1TX=(T(I+1,J,K)-T(I-1,J,K))/(2*DX)
23     D1TY=(T(I,J+1,K)-T(I,J-1,K))/(2*DY)
24     D1TWZ=(T(I,J,K+1)*W(I,J,K+1)-T(I,J,K-1)*W(I,J,K-1))/(2*DZ)
25     D2TX=(T(I+1,J,K)+T(I-1,J,K)-2*T(I,J,K))/(DX*DX)
26     D2TY=(T(I,J+1,K)+T(I,J-1,K)-2*T(I,J,K))/(DY*DY)
27     D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
28     D1TZ=(T(I,J,K+1)-T(I,J,K-1))/(2*DZ)
29     IF(MAR(I,J).EQ.11) GO TO 200
30     D1TWZ=0.0
31     200 CONTINUE
32     IF (K.EQ.1) GO TO 24
33     IF (K.EQ.KN) GO TO 20
34     GO TO 21
35     20 CONTINUE
36     GO TO 30
37     D1TWZ=0.0
38     D2TZ=2*(T(I,J,K-1)-T(I,J,K)+CB*HI(I,J)*DZ)/(DZ*DZ)
39     D1TZ=0.0
40     GO TO 21
41     24 CONTINUE
42     CT=AKT*((T(I,J,1)*TREF+TREF)-TAMB)/TREF
43     CT=CT*HI(I,J)
44     D1TZ=CT
45     D1TWZ=(4.*T(I,J,K+1)*W(I,J,K+1)-3.*T(I,J,K)*W(I,J,K)-T(I,J,K+2)
46     C*W(I,J,K+2))/(2.*DZ)
47     D2TZ=2*(T(I,J,2)-CT*DZ-T(I,J,1))/(DZ*DZ)
48     21 CONTINUE
49     TD(I,J,K)=(1.C/HI(I,J))*(-TAI*(D1HTUX+D1HTVY+HI(I,J)*D1TWZ)
50     C+TAH*(D1TX+HX(I,J)+D2TX*HI(I,J)+D1TY*HY(I,J)+D2TY*HI(I,J))
51     C+(TAV/HI(I,J))*(B3+D2TZ+D1TZ*D1B3Z))+DT*T(I,J,K)
52     GO TO 6
53     30 CONTINUE
54     TD(I,J,KN)=T(I,J,KN)
55     8 CONTINUE
56     9 CONTINUE

```



57	10	CONTINUE
53		RETURN
59		END

PRECEDING PAGE BLANK NOT FILMED.

## 9.2.62 TEQB

This program allows for vertical mixing at a particular grid point, if the temperature of the fluid at the grid point just above it is less and the difference of temperatures is more than a specified maximum, then the temperatures at the two points are averaged.

\*DOC.TEGB

```

1  SUBROUTINE TECB(I,J,K,IN,JN,KN,T,MAR)
2  DIMENSION T(IN,JN,KN),MAR(IN,JN)
3  KMM1=KN-1
4  DO 10 I=1,IN
5  DO 10 J=1,JN
6  IF (MAR(I,J).EQ.0) GO TO 9
7  110 CONTINUE
8  MARK=0
9  DO 8 K=1,KMM1
10  TX=C.CG2
11  TT=T(I,J,K)+TX
12  IF (K.EQ.1) GO TO 7
13  IF (K.EQ.KMM1) GO TO 6
14  IF (TT.GE.T(I,J,K+1)) GO TO 111
15  MARK=1
16  AVT=(TT+T(I,J,K+1))/2.0
17  T(I,J,K)=AVT-TX
18  T(I,J,K+1)=AVT
19  GO TO 5
20  7 CONTINUE
21  IF (TT.GE.T(I,J,K+1)) GO TO 111
22  MARK=1
23  AVT=(TT+2*T(I,J,K+1))/3.0
24  T(I,J,K)=AVT-TX
25  T(I,J,K+1)=AVT
26  GO TO 5
27  6 CONTINUE
28  IF (TT.GE.T(I,J,K+1)) GO TO 111
29  MARK=1
30  AVT=(2*TT+T(I,J,K+1))/3.0
31  T(I,J,K)=AVT-TX
32  T(I,J,K+1)=AVT
33  GO TO 5
34  5 CONTINUE
35  111 CONTINUE
36  8 CONTINUE
37  9 CONTINUE
38  10 CONTINUE
39  RETURN
40  END

```

## 9.2.63 TPRIN1

This program prints the input parameters.

\*DOC.TPRIN1

```

1  SUBROUTINE TPRIN1(TAI,TAH,TAV,CB,CW,AKT,TREF,RREF,EUL,A,B,C,TE,TO)
2  PRINT 10, TAI,TAH,TAV,CB,CW,AKT,TREF,RREF,EUL,A,B,C,TE,TO
3      10  FORMAT (/' TAI=',E15.7,/' TAH=',E15.7,/' TAV=',E15.7,/' CB=',E15.7
4  C,/' CW=',E15.7,/' AKT=',E15.7,/' TREF=',E15.7,/' RREF=',E15.7,/'
5  C' EUL=',E15.7,/' A=',E15.7,/' B=',E15.7,/' C=',E15.7,/' TE=',E15.7
6  C,/' TO=',E15.7,/'
7  RETURN
8  END

```

9.2.64 TPRIN2

This program prints temperature and density fields.

\*DOC.TPRIN2

```

1 C*****
2 C      THIS PROGRAM PRINTS THE VALUES OF T,TW,RO,ROW
3 C*****
4      SUBROUTINE TPRIN2(I,J,K,IN,JN,KN,T,RO,TREF)
5      DIMENSION T(IN,JN,KN),RO(IN,JN,KN)
6      DO 10 K=1,KN
7      DO 100 I=1,IN
8      PRINT 11,K,I,(T(I,J,K),J=1,JN)
9      10 PRINT 12,(RO(I,J,K),J=1,JN)
10     11 FORMAT(/' K=' ,I3,'X',' I=' ,I3/' TEMPERATURE'/(5X,8E15.7))
11     12 FORMAT(' DENSITY'/(5X,8E15.7))
12     DO 100 K=1,KN
13     DO 100 J=1,JN
14     DO 100 I=1,IN
15     100 T(I,J,K)=(1.+T(I,J,K))*TREF
16     DO 150 K=1,KN
17     WRITE(6,105) K
18     DO 140 I=1,IN
19     WRITE(6,106) (T(I,J,K),J=1,JN)
20     140 CONTINUE
21     150 CONTINUE
22     105 FORMAT('1', ' TEMPERATURES FOR K=',I5)
23     106 FORMAT('/',22F5.1)
24     RETURN
25     END

```

U.S. GOVERNMENT  
 PRINTING OFFICE

## 9.2.65 TPRIN9

This subroutine is used by the far field models. It prints temperature and density fields.



.H=DULL(1).TPRIN9

```

1 C*****
2 C      THIS PROGRAM PRINTS THE VALUES OF T,IN,RO,POW
3 C*****
4 SUBROUTINE TPRIN9(I,J,K,IN,JN,KN,T,RO,TREF,MAR)
5 DIMENSION T(IN,JN,KN),RO(IN,JN,KN),MAR(IN,JN)
6 GO TO 512
7 DO 10 K=1,KN
8 DO 10 I=1,IN
9 PRINT 11,K,I,(T(I,J,K),J=1,JN)
10 10 PRINT 12,(RO(I,J,K),J=1,JN)
11 11 FORMAT(/' K=',I3,'X',' I=',I3/' TEMPERATURE'/(EX,2E15.7))
12 12 FORMAT(' DENSITY'/(EX,2E15.7))
13 512 CONTINUE
14 DO 100 K=1,KN
15 DO 100 J=1,JN
16 DO 100 I=1,IN
17 T(I,J,K)=(1.+T(I,J,K))*TREF
18 IF(MAR(I,J).EQ.C) T(I,J,K)=1000000.CC
19 100 CONTINUE
20 DO 150 K=1,KN
21 WRITE(6,105) K
22 DO 140 I=1,IN
23 WRITE(6,106) (T(I,J,K),J=1,JN)
24 140 CONTINUE
25 150 CONTINUE
26 105 FORMAT('1',,, ' TEMPERATURES FOR K=',I5,/)
27 106 FORMAT(//,18F7.1)
28 RETURN
29 END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

#### 9.2.66 TPRINA

This subroutine is used by the far field printing programs. The subroutine is similar to TPRIN2 with the difference that it is tailored to the Lake Belcws application.

M=DULL(1).TPRINA

```

1  C*****
2  C      THIS PROGRAM PRINTS THE VALUES OF T,TW,RC,RO
3  C*****
4      SUBROUTINE TPRINA(I,J,K,IN,JN,KN,T,FO,TREF,MAR)
5      DIMENSION T(I,J,KN),RO(I,J,KN),MAR(I,J,N)
6      DO 10 K=1,KN
7      DO 10 I=1,IN
8      PRINT 11,K,I,(T(I,J,K),J=1,JN)
9      PRINT 12,(RO(I,J,K),J=1,JN)
10     FORMAT(' K=',I',IX,'I=',I',/ TEMPERATURE'/(IX,8E15.7))
11     FORMAT(' DENSITY'/(5X,8E15.7))
12     DO 100 K=1,KN
13     DO 100 J=1,JN
14     DO 100 I=1,IN
15     T(I,J,K)=(1.+T(I,J,K))*TREF
16     IF(MAR(I,J).EQ.0) T(I,J,K)=1000000.00
17     100 CONTINUE
18     DO 150 K=1,KN
19     WRITE(6,100) K
20     DO 140 I=1,IN
21     WRITE(6,100) (T(I,J,K),J=1,JN)
22     140 CONTINUE
23     150 CONTINUE
24     105 FORMAT('1',,,* TEMPERATURES FOR K=',I5,/)
25     106 FORMAT(/,18F7.1)
26     RETURN
27     END

```

## 92.67 UANVC

This subroutine adds coriolis component to H and G which are obtained from subroutine "UV"

10DOC.UANVC

```

1      SUBROUTINE UANVC(I,J,K,IN,JN,KN,ABR,DT,U,V,H,G,HI,MAR)
2      DIMENSION H(IN,JN,KN),G(IN,JN,KN),U(IN,JN,KN),V(IN,JN,KN),
3      CHI(IN,JN),MAR(IN,JN)
4      KNM1=KN-1
5      DO 10 I=1,IN
6      DO 10 J=1,JN
7      IF (MAR(I,J).LT.11) GO TO 9
8      DO 8 K=2,KNM1
9      H(I,J,K)=H(I,J,K)+ABR*HI(I,J)*V(I,J,K)*DT
10     G(I,J,K)=G(I,J,K)-ABR*HI(I,J)*U(I,J,K)*DT
11     CONTINUE
12     CONTINUE
13     CONTINUE
14     RETURN
15     END

```

## 9.2.68 UV

This subroutine computes new horizontal velocities without including the Coriolis component which is added later in the subroutine UANVC. This program uses U and V momentum equations and computes velocities in the interior points after one time step and are stored as H and G.

\*DOC.UV

```

1  SUBROUTINE UV(I,J,K,IM,JW,IN,JN,KN,IMN,JWN,U,V,D,E,H,G,DX,DY,DZ,
2  CM,
3  CDT,AI,AP,AH,AV,AJ,HI,HX,HY,P,MAR)
4  DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
5  CH(IN,JN,KN),G(IN,JN,KN),HI(IN,JN),HX(IN,JN),HY(IN,JN),P(IMN,JWN),
6  CHAR(IM,JN)
7  DIMENSION W(IN,JN,KN)
8  DIMENSION A3(KN)
9  KNH1=KN-1
10 A=DT*AH*(1/(DX*CX)+1/(DY*DY))
11 DO 10 I=1,IN
12 DO 10 J=1,JN
13   IM=I
14   JW=J
15   IF (MAR(I,J).LT.11) GO TO 9
16   D1PX=(P(IM,JW)-P(IM-1,JW)+P(IM,JW-1)-P(IM-1,JW-1))/(2*DX)
17   D1PY=(P(IM,JW)-P(IM,JW-1)+P(IM-1,JW)-P(IM-1,JW-1))/(2*DY)
18   DO 8 K=2,KNH1
19   B=DT*AV*A3(K)/(CZ*DZ)
20   C=(HI(I,J)+A*HI(I,J)+B/HI(I,J))
21   D1HUUX=(U(I+1,J,K)+U(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
22   C*U(I-1,J,K)+HI(I-1,J))/(2*DX)
23   D1HUVY=(U(I,J+1,K)+V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
24   C*V(I,J-1,K)+HI(I,J-1))/(2*DY)
25   D1HUVX=(U(I+1,J,K)+V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
26   C*V(I-1,J,K)+HI(I-1,J))/(2*DX)
27   D1HVVY=(V(I,J+1,K)+V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
28   CV(I,J-1,K)+HI(I,J-1))/(2*DY)
29   D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
30   D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
31   D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
32   D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
33   D1UWZ=(U(I,J,K+1)+W(I,J,K+1)-U(I,J,K-1)+W(I,J,K-1))/(2*DZ)
34   D1VWZ=(V(I,J,K+1)+W(I,J,K+1)-V(I,J,K-1)+W(I,J,K-1))/(2*DZ)
35   DDUX=(U(I+1,J,K)+U(I-1,J,K)-D(I,J,K))/(DX*DX)
36   DDUY=(U(I,J+1,K)+U(I,J-1,K)-D(I,J,K))/(DY*DY)
37   DCUZ=(U(I,J,K+1)+U(I,J,K-1)-D(I,J,K))/(DZ*DZ)
38   DDVX=(V(I+1,J,K)+V(I-1,J,K)-E(I,J,K))/(DX*DX)
39   DDVY=(V(I,J+1,K)+V(I,J-1,K)-E(I,J,K))/(DY*DY)
40   DDVZ=(V(I,J,K+1)+V(I,J,K-1)-E(I,J,K))/(DZ*DZ)
41   H(I,J,K)=(DT/C)*(-AI*(D1HUUX+D1HUVY+HI(I,J)+D1UWZ)-HI(I,J)*AP
42   C*D1PX+AH*HI(I,J)*(DDUX+DDUY)+AH*HX(I,J)*D1UX+AH*HY(I,J)*D1UY
43   C+AV*A3(K)*DDUZ/HI(I,J)+HI(I,J)*U(I,J,K)/C
44   G(I,J,K)=(DT/C)*(-AI*(D1HUVX+D1HVVY+HI(I,J)+D1VWZ)-HI(I,J)*AP
45   C*D1PY+AH*HI(I,J)*(DDVX+DDVY)+AH*HX(I,J)*D1VX+AH*HY(I,J)*D1VY
46   C+AV*A3(K)*DDVZ/HI(I,J)+HI(I,J)*V(I,J,K)/C
47   8 CONTINUE
48   9 CONTINUE
49   10 CONTINUE
50   RETURN
51   END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

## 9.2.69 UVT

This program is called in by TMAIN. This program computes U and V for variable density model, at successive time steps.  $\alpha$  and  $\beta$  momentum equations are used for computations. Results are stored as H (for U) and G (for V).



DOOC.UVT

```

SUBROUTINE UVT(I,J,K,IM,JM,IN,JN,KN,IMN,JMN,U,V,D,E,H,G,DX,DY,DZ,
CRINTX,CRINTY,EUL,
CN,
COT,A1,AP,AH,AV,A3,HI,HX,HY,P,HAR)
  DIMENSION U(IM,JN,KN),V(IM,JN,KN),D(IM,JN,KN),E(IM,JN,KN),
  CH(IM,JN,KN),G(IM,JN,KN),HI(IM,JN),HX(IM,JN),HY(IM,JN),P(IMN,JMN),
  CHAR(IM,JN)
  DIMENSION M(IM,JN,KN)
  DIMENSION A3(KN)
  DIMENSION CRINTX(IM,JN,KN),CRINTY(IM,JN,KN)
  KNM1=KN-1
  A=OT*AH*(1/(DX*CX)+1/(DY*DY))
  DO 1C I=1,IM
  DO 1C J=1,JN
  I=I
  JM=J
  IF (HAR(I,J).LT.11) GO TO 9
  DO 8 K=2,KNM1
  DIPX=(P(IM,JM)-P(IM-1,JM)+P(IM,JM-1)-P(IM-1,JM-1))/(2*DX)
  C+EUL*CRINTX(I,J,K)
  DIPY=(P(IM,JM)-P(IM,JM-1)+P(IM-1,JM)-P(IM-1,JM-1))/(2*DY)
  C+EUL*CRINTY(I,J,K)
  B=OT*AV*A3(K)/(CZ*DZ)
  C=(HI(I,J)+A*HI(I,J)+B/HI(I,J))
  DIHUUX=(U(I+1,J,K)+U(I+1,J,K)+HI(I+1,J)-U(I-1,J,K)
  C+U(I-1,J,K)+HI(I-1,J))/(2*DX)
  DIHUVY=(U(I,J+1,K)+V(I,J+1,K)+HI(I,J+1)-U(I,J-1,K)
  C+V(I,J-1,K)+HI(I,J-1))/(2*DY)
  DIHUVX=(U(I+1,J,K)+V(I+1,J,K)+HI(I+1,J)-U(I-1,J,K)
  C+V(I-1,J,K)+HI(I-1,J))/(2*DX)
  DIHVYV=(V(I,J+1,K)+V(I,J+1,K)+HI(I,J+1)-V(I,J-1,K)+
  C+V(I,J-1,K)+HI(I,J-1))/(2*DY)
  DIUX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
  DIUY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
  DIVX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
  DIVY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
  DIUMZ=(U(I,J,K+1)+W(I,J,K+1)-U(I,J,K-1)+W(I,J,K-1))/(2*DZ)
  DIVMZ=(V(I,J,K+1)+W(I,J,K+1)-V(I,J,K-1)+W(I,J,K-1))/(2*DZ)
  DDUX=(U(I+1,J,K)+U(I-1,J,K)-D(I,J,K))/(DX*DX)
  DDUY=(U(I,J+1,K)+U(I,J-1,K)-D(I,J,K))/(DY*DY)
  DDUZ=(U(I,J,K+1)+U(I,J,K-1)-D(I,J,K))/(DZ*DZ)
  DDVX=(V(I+1,J,K)+V(I-1,J,K)-E(I,J,K))/(DX*DX)
  DDVY=(V(I,J+1,K)+V(I,J-1,K)-E(I,J,K))/(DY*DY)
  DDVZ=(V(I,J,K+1)+V(I,J,K-1)-E(I,J,K))/(DZ*DZ)
  HI(I,J,K)=(DT/C)*(-AI*(DIHUUX+DIHUVY+HI(I,J)+DIUMZ)-HI(I,J)*AP
  C*DIPX+AH*HI(I,J)*(DDUX+DDUY)+AH*HX(I,J)+DIUX+AH*HY(I,J)+DIUY
  C+AV*A3(K)+DDUZ/HI(I,J))+HI(I,J)+U(I,J,K)/C
  G(I,J,K)=(OT/C)*(-AI*(DIHUVX+DIHVYV+HI(I,J)+DIVMZ)-HI(I,J)*AP
  C*DIPY+AH*HI(I,J)*(DDVX+DDVY)+AH*HX(I,J)+DIVX+AH*HY(I,J)+DIVY
  C+AV*A3(K)+DDVZ/HI(I,J))+HI(I,J)+V(I,J,K)/C
  CONTINUE
  CONTINUE
  CONTINUE
  RETURN
  END

```

8  
9  
1C

## 9.2.70 UVTB

This subroutine is used by the far field stratified model. The subroutine is similar to UVT with the difference that it calls for subroutine VERTDF which supplies the value for vertical viscosity and its derivative.

M\*DJLL(1).UVTB

```

1  SUBROUTINE UVTB(I,J,K,IX,IY,IZ,IN,JN,KN,IWN,JWN,U,V,C,E,H,G,DX,DY,DZ,
2  CRINTX,CRINTY,LUL,
3  CW,
4  CDT,AT,AP,AH,AV,AZ,HI,HX,HY,P,MAR,T,TREF,CONS,AVMX,AVMN)
5  DIMENSION U(IN,JN,KN),V(IN,JN,KN),C(IN,JN,KN),E(IN,JN,KN),
6  CH(IN,JN,KN),G(IN,JN,KN),HI(IN,JN),HX(IN,JN),HY(IN,JN),P(IWN,JWN),
7  CHAR(IN,JN)
8  DIMENSION A(IN,JN,KN),T(IN,JN,KN)
9  DIMENSION A3(KN)
10 DIMENSION RINTX(IN,JN,KN),RINTY(IN,JN,KN)
11 KNM1=KN-1
12 A=DT*AH*(1/(DX*DX)+1/(DY*DY))
13 DO 10 I=1,IN
14 DO 10 J=1,JN
15 IW=I
16 JW=J
17 IF (MAR(I,J).LT.11) GO TO 9
18 DO 9 K=2,KNM1
19 DIPX=(P(IW,JW)-P(IW-1,JW)+P(IW,JW-1)-P(IW-1,JW-1))/(2*DX)
20 C+EUL*RINTX(I,J,K)
21 DIPY=(P(IW,JW)-P(IW,JW-1)+P(IW-1,JW)-P(IW-1,JW-1))/(2*DY)
22 C+EUL*RINTY(I,J,K)
23 CALL VERIDF(I,J,K,IN,JN,KN,HI,AB3,D1A3Z,D1B3Z,D7,T,A3,TREF
24 C,CONS,AVMX,AVMN)
25 A3(K)=AB3
26 B=DT*AV*A3(K)/(DZ*DZ)
27 C=(HI(I,J)+A*HI(I,J)+B/HI(I,J))
28 DIHUX=(U(I+1,J,K)+U(I-1,J,K)*HI(I+1,J)-U(I-1,J,K)
29 C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
30 DIHUY=(U(I,J+1,K)+V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
31 C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
32 DIHVV=(U(I+1,J,K)+V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
33 C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
34 DIHVVY=(V(I,J+1,K)+V(I,J-1,K)*HI(I,J+1)-V(I,J-1,K)*
35 CV(I,J-1,K)*HI(I,J-1))/(2*DY)
36 DIUWZ=(U(I,J,K+1)-U(I,J,K-1))/(2*DZ)
37 DIVZ=(V(I,J,K+1)-V(I,J,K-1))/(2*DZ)
38 DIUX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
39 DIUY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
40 DIVX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
41 DIVY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
42 DIUWZ=(U(I,J,K+1)*HI(I,J,K+1)-U(I,J,K-1)*HI(I,J,K-1))/(2*DZ)
43 DIVWZ=(V(I,J,K+1)*HI(I,J,K+1)-V(I,J,K-1)*HI(I,J,K-1))/(2*DZ)
44 DDUX=(U(I+1,J,K)+U(I-1,J,K)-C(I,J,K))/(DX*DX)
45 DDUY=(U(I,J+1,K)+U(I,J-1,K)-C(I,J,K))/(DY*DY)
46 DDUZ=(U(I,J,K+1)+U(I,J,K-1)-C(I,J,K))/(DZ*DZ)
47 DDVX=(V(I+1,J,K)+V(I-1,J,K)-E(I,J,K))/(DX*DX)
48 DDVY=(V(I,J+1,K)+V(I,J-1,K)-E(I,J,K))/(DY*DY)
49 DDVZ=(V(I,J,K+1)+V(I,J,K-1)-E(I,J,K))/(DZ*DZ)
50 H(I,J,K)=(CDT/C)*(1-AT*(DIHUX+DIHUY+HI(I,J)*DIUWZ)-HI(I,J)*P
51 C*DIPX+AH*HI(I,J)*(DDUX+DDUY)+AH*HX(I,J)*DIUX+AH*HY(I,J)*DIUY
52 C+AV*A3(K)*DDUZ+D1B3Z*D1UT/HI(I,J)+PI(I,J)*DIY(I,J)/C
53 G(I,J,K)=(DT/C)*(1-AT*(DIHVV+DIHVVY+HI(I,J)*DIWZ)-HI(I,J)*P
54 C*DIPY+AH*HI(I,J)*(DDVX+DDVY)+AH*HY(I,J)*DIVX+AH*HY(I,J)*DIY
55 C+AV*A3(K)*DDVZ+D1B3Z*D1VZ/HI(I,J)+HI(I,J)*V(I,J,K)/C
56 CONTINUE

```

8

OF POOR QUALITY

57	9	CONTINUE
58	10	CONTINUE
59		RETURN
60		END

PRECEDING PAGE BLANK NOT FILMED

9.2.71 UVTOP

This program computes U and V at the top using wind stress boundary conditions. Computations are made for MAR < 11 only. (Interior grid points).

\*DOC.UVTOP

```

1 C*****
2 C      THIS PROGRAM CALCULATES U AND V VELOCITIES AT THE SURFACE USING STRESS
3 C      BOUNDARY CONDITIONS
4 C*****
5      SUBROUTINE UVTOP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
6      DIMENSION H(IN,JN,KN),G(IN,JN,KN)
7      DIMENSION HI(IN,JN),MAR(IN,JN)
8      DO 800 I=1,IN
9      DO 800 J=1,JN
10     IF (MAR(I,J).LT.11) GO TO 700
11     K=1
12     TX=TAUX*HI(I,J)
13     TY=TAUY*HI(I,J)
14     H(I,J,K)=(4*H(I,J,K+1)-H(I,J,K+2)-2*DZ*TX)/3
15     G(I,J,K)=(4*G(I,J,K+1)-G(I,J,K+2)-2*DZ*TY)/3
16     700 CONTINUE
17     800 CONTINUE
18     RETURN
19     END

```

## 9.2.72 VERTDF

This program compute vertical viscosity and diffusivity as a function of temperature gradient in the vertical direction. The program also computes first derivative of vertical eddy viscosity and diffusivity with respect to  $x$ .

M=DULL(1).VERTOF

```

1      SUBROUTINE VERTOF (I,J,K,IN,JN,KN,HI,A3,D1A3Z,D1B3Z,DZ,T,A3,TRCF
2      C,CONS,AVMX,AVMN)
3      DIMENSION HI(IN,JN),T(IN,JN,KN),A3(KI)
4      DO 50 KK=1,KN
5      IF(KK.EQ.1) GO TO 11
6      IF(KK.EQ.KN) GO TO 12
7      D1T2=(T(I,J,KN+1)-T(I,J,KN-1))/(2.*DZ)
8      GO TO 20
9      11 D1T2=(4.*T(I,J,KN+1)-3.*T(I,J,KN)-T(I,J,KN-1))/(2.*DZ)
10     GO TO 20
11     12 D1T2=(-4.*T(I,J,KN-1)+3.*T(I,J,KN)+T(I,J,KN-2))/(2.*DZ)
12     20 PARA=-HI(I,J)*(((KN-1)*DZ)**2)*D1T2
13     A3(KK)=AVMX/(1.+PARA*CONS)
14     IF(P3(KK).LT.AVMN) A3(KK)=AVMN
15     DEPTH=(FLOAT(KK-1))*DZ*HI(I,J)
16     IF(DEPTH.GT.0.30) A3(KK)=AVMN
17     50 CONTINUE
18     IF(K.EQ.1) D1A3Z=0.0
19     IF(K.EQ.KN) D1A3Z=(-4.*A3(K-1)+3.*A3(K)+A3(K-2))/(2.*DZ)
20     IF(K.GT.1.AND.K.LT.KN) D1A3Z=(A3(K+1)-A3(K-1))/(2.*DZ)
21     A3=A3(K)
22     D1B3Z=D1A3Z
23     IF(I.EQ.24.AND.J.EQ.6) WRITE(6,100) I,J,K,A3,D1A3Z,D1B3Z
24     100 FORMAT (3I5,3F15.0)
25     RETURN
26     END
RT,5 J.WHTOP

```

ORIGINAL PAGE IS  
OF POOR QUALITY



## 9.2.73 WHATIJ

This program computes the values of  $W$  at  $I, J$  from the values of  $WH$  at  $IW, JW$ .

Vertical velocity at a point on the main grid point is taken to be equal to the average of vertical velocities at four points on the half grid and lying adjacent to the point under consideration. Computations are made for interior grid points only.

DOC.WHATIJ

```

1 C*****
2 C   THIS PROGRAM CALCULATES THE VALUE OF W AT I,J FROM VALUES OF WH AT IW,JW
3 C*****
4   SUBROUTINE WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
5   DIMENSION WH(IWN,JWN,KN),W(IN,JN,KN)
6   DIMENSION MAR(IN,JN)
7   DO 3550 I=1,IN
8   DO 3550 J=1,JN
9   IF (MAR(I,J).LT.11) GO TO 3540
10  DO 3510 K=1,KN
11  IW=I
12  JW=J
13  W(I,J,K)=(WH(IW,JW,K)+WH(IW,JW-1,K)+WH(IW-1,JW,K)+WH(IW-1,JW-1,K))
14  C/4.
15  3510 CONTINUE
16  3540 CONTINUE
17  3550 CONTINUE
18  RETURN
19  END

```

## 9.2.74 WHTOP

This program sets the value of WH equal to zero at the surface.

$$WH = \frac{\partial \gamma}{\partial t} = 0 \text{ at } x=0$$

No computations are made for points outside of the region of interest, defined by MRH = 0.

```

*DOC.WHTOP
1 C.....
2 C THIS PROGRAM SETS THE VALUE OF WH EQUAL TO ZERO AT THE SURFACE
3 C.....
4 SUBROUTINE WHTOP(IW,JW,IWN,JWN,KN,WH,K,MRH)
5 DIMENSION WH(IWN,JWN,KN)
6 DIMENSION MRH(IWN,JWN)
7 DO 3300 IW=1,IWN
8 DO 3300 JW=1,JWN
9 IF (MRH(IW,JW).EQ.0) GO TO 3000
10 WH(IW,JW,1)=0
11 3000 CONTINUE
12 3300 CONTINUE
13 RETURN
14 END

```

ORIGINAL PAGE IS  
OF POOR QUALITY